

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

STUDY OF HIGH DENSITY E. TRANSITION REQUIREMENTS VERSUS THE EFFECTS ON BCH ERROR CORRECTING CODE

(NASA-CR-170643) HIGH DENSITY BIT
TRANSITION REQUIREMENTS VERSUS THE EFFECTS
ON BCH ERROR CORRECTING CODE Interim Report
(Mississippi State Univ.) 130 p
HC A07/MF A01

N83-11784

Unclass

CSCL 09D G3/61 01024



ENGINEERING & INDUSTRIAL RESEARCH STATION
ELECTRICAL ENGINEERING

INTERIM REPORT

Submitted to:

**George C. Marshall Space Flight Center
National Aeronautics and Space Administration**



Principal Investigator:

✓ **Frank M. Ingels**

Associate Investigator:

William O. Schoggen

Mississippi State University
Mississippi State, MS 39762

MSSU-EIRS-EE-82-4

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BHC
ERROR CORRECTING CODING

<u>ITEM</u>	<u>PAGE</u>
SUMMARY	1
APPENDIX A REDUCING THE BIT ERROR RATE OF A DATA STREAM . .	2
APPENDIX B PROBABILITY OF ACHIEVING FRAME SYNCHRONIZATION .	7
APPENDIX C REPORT OF PDR (MARTIN/DENVER)	17
APPENDIX D REPORT OF CDR (MBB/MUNICH)	21
APPENDIX E INTERIM FINAL REPORT JULY L, 1980 to DECEMBER 31, 1981	28

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

SUMMARY

This report constitutes the final fulfillment of the requirements of the contract NAS8-33887. A comprehensive technical interim final report was submitted in January 1982 and an interim report was submitted on September 16, 1980.

The interim report of September 16, 1980 contained the recommended design to achieve the required bit transition density for the HRM data stream of the Space Laboratory Vehicle. It contained a recommended circuit approach, specified the pseudo random (PN) sequence to be used and detailed the properties of the sequence. Calculations showing the probability of failing to meet the required transition density were included in this report.

The technical interim final report of January 1982 included a computer simulation of the data stream & 100% cover sequence. All worst case situations were simulated and the bit transition density always exceeded that required. An interested reader is referred to that comprehensive report (MSU report MSSU-EIRS-EE-81-5, Appendix E).

In March 1982 the Preliminary Design Review (PDR) was held at Martin Marietta/Denver and the critical Design Review (CDR) was held at MBB in Munich, Germany in June 1982. This investigator attended both meetings and reports have been filed for both of these activities. These reports are included as appendices of this final report.

Several related activities were requested of this investigator and the reports concerning these activities are also included as appendices of this final report.

In summary it should be noted that the CSG Encoder/Decoder design has been constructed and demonstrated by MBB and Martin Marietta/Denver. The demonstrations were successful and all HRM and HRDM units will contain the CSG encoder or CSG decoder as appropriate.

APPENDIX A

REDUCING THE BIT ERROR RATE OF A DATA STREAM

A request was made to suggest a method that would be inexpensive and yet effective for achieving an average bit error rate of 10^{-6} from a data stream with an average bit error rate (BER) of 10^{-5} . The following material of appendix A is the response to this request.

MISSISSIPPI STATE UNIVERSITY
COLLEGE OF ENGINEERING



DEPARTMENT OF ELECTRICAL ENGINEERING
DRAWER EE
MISSISSIPPI STATE, MISSISSIPPI 39762
PHONE (601) 325-3912

December 17, 1980

Mr. Russ Coffee
EF13
MSFC, Alabama 35812

Dear Mr. Coffee:

Enclosed is a write up concerning a method of effectively lowering the average BER from 10^{-5} to 10^{-6} for additive white gaussian noise.

This is only one possible method of many and it is to be stressed that a detailed knowledge of the structure of the frame of data is necessary to properly match the data source to the coding technique.

If I can be of any other help please contact me.

Sincerely,

Frank Ingels
Professor

On Lowering the Average Bit Rate From 10^{-5} to Less than 10^{-6}

1. Assumptions 10^{-5} average bit error rate due to additive white gaussian noise.
2. Assume that 1% or 2% overhead may be added to data stream by simply increasing number of words in frame and upping output data stream rate by 1% or 2%.
3. Assume frame is roughly in multiples of 1000 bits.
(actually 1013 or less)

Structuring Frame in 1013 segments we may add 10 extra digits per segment thereby producing 1023 bit segments. Thus we have a (1023, 1013, 1) BCH code which has 1013 information digits (actually 63 words, 16 bits each, will be 1008 digits and we have 5 unused digits. Hence the overhead will actually be 15 digits out of 1008 for 1.488% overhead. Thus if the frame consist of 63 words, 126 words or multiple of 63 words we may add error correcting for 1 digit out of 1000 for an overhead of approximately 1.5%).

Probability of error in 1013 digits unencoded is:

$$\begin{aligned} P(\text{error, unencoded}) &= 1 - P(0 \text{ errors in } 1013 \text{ digits}) \\ &= 1 - (.99999)^{1013} = 1.0078914 \cdot 10^{-2} = 1.0078914\% \end{aligned}$$

For the (1023,1013,1) code the probability of error in 1023 digits is:

$$\begin{aligned} P(\text{error, encoded}) &= 1 - [P(0 \text{ errors in } 1023) + P(1 \text{ error in } 1023)] \\ &= 1 - [(.99999)^{1023} + 1023(.99999)^{1022}(10^{-5})] \\ &= 5.19208 \times 10^{-5} = .00519208\% \end{aligned}$$

The equivalent average bit error rate due to additive gaussian noise would thus be

$$[1 - P(0 \text{ errors for } 1023)] = (1 - X^{1023}) = 5.19208 \cdot 10^{-5}$$

or

$$X^{1023} = .99994807$$

or

$$1023(\text{Log}_{10} X = \text{Log}_{10}(.99994807))$$

$$X = .99999994$$

$$\text{of Average Eq Bit Error Rate} = 1 - X = 5.05 \times 10^{-8}$$

Thus a significant reduction in equivalent bit error rate is achieved through this simple encoding.

The encoder is shown in Figure 1. The decoder may be an encoder circuit plus a 1023×1013 (or 10^6 digit) ROM or the error trapping decoder shown in Figure 2. The tradeoff is 1023 clock pulses versus 10^6 bits of ROM.

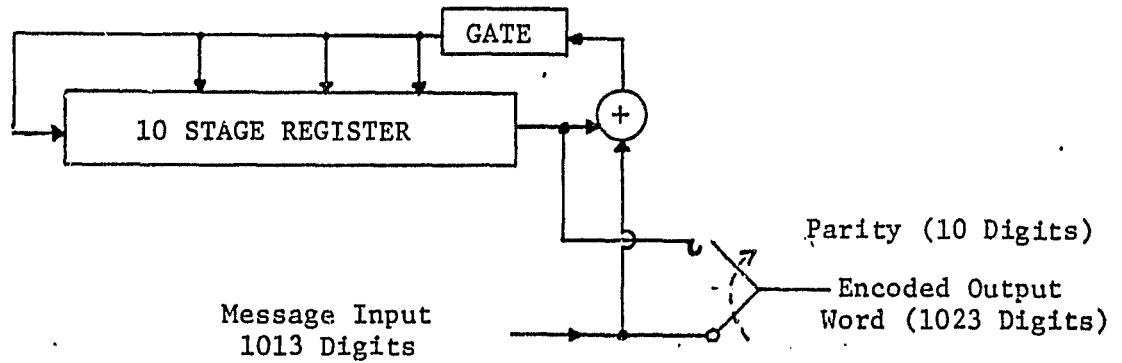


Figure 1. Encoder (1023,1013,1) BCH Code

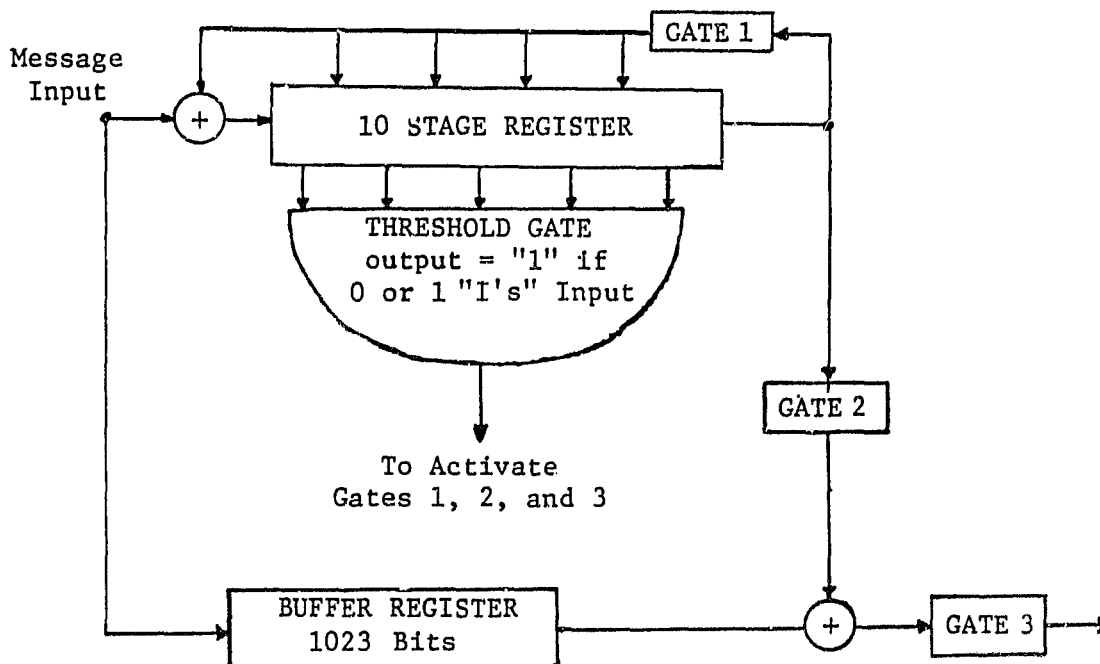


Figure 2. Error Trapping Decoder for (1023,1013,1) BCH Code

APPENDIX B

PROBABILITY OF ACHIEVING FRAME SYNCHRONIZATION

A request was made to conduct an analysis of the probability of achieving frame synchronization (sync) within 3 frames of data with and without a 1 bit per frame sync allowable error. The analysis was conducted and the report follows as Appendix B.

MISSISSIPPI STATE UNIVERSITY
COLLEGE OF ENGINEERING



DEPARTMENT OF ELECTRICAL ENGINEERING
DRAWER EE
MISSISSIPPI STATE, MISSISSIPPI 39762
PHONE (601) 325-3312

March 15, 1982

Mr. Russ Coffey
EF-13
MSFC, AL 35812

Dear Mr. Coffey:

Enclosed is an independent analysis of the probability of achieving frame sync in 3 frames of data. You may notice that I have deemed it necessary to include the possibility of errors occurring in the frame count word when detecting the first frame sync word. This is necessary since the second detection of a correct frame sync word (when in Verify Mode) will automatically check to see if the frame count is correctly incrementing (or decrementing if encoded with the PN sequence) with respect to the first correctly determined frame sync word. Thus, an error in the first frame count will result in non-verification in the second frame count even if it is correct.

If you have any questions, please call.

Sincerely,

A handwritten signature in cursive script that reads 'Frank Ingels'.

Frank Ingels, Ph.D.

Professor

Department of Electrical Engineering

lan

Enclosures

cc: Mr. Frank Echols

PROBABILITY OF FRAME SYNCHRONIZATION LOCK WITHIN
3 FRAMES FOR HRDM 2 MHZ DATA STREAM

ANALYSIS

Assumptions:

1. It is assumed that zero (0) errors are allowed in a 28 bit frame synchronization (sync) word and that (0) errors are allowed in the frame count word for recognition to occur.
2. To enter the Verification Mode (VM), from the Search Mode (SM) it is required that the 28 bit frame sync word be detected with zero errors.
3. To enter the Lock Mode (LM) from VM, it is required that the 28 bit frame sync word be detected in the next successive time location with no errors, and that the frame count has incremented (or decremented) by 1 count with respect to the first detected frame sync frame count.
4. The frame sync 28 bit pattern in question is:

1111 1010 1111 0011 0011 0100 0000 4 Bit ID.

Let the following terms be defined:

- P_o = Probability of zero errors in the 28 bit frame sync pattern.
- P_{FC} = Probability of zero errors in the 4 bit frame count pattern.
- P_{FS} = Probability of a false sync word occurring in the data stream or in the slipped frame sync pattern.

P_{VM} = Probability of entering the Verification Mode from the
Search Mode

P_{LM} = Probability of entering the Lock Mode from the
Verification Mode.

To determine the probability of sync Lock, P_{3F} , in 3 frames of data we first note the following: For 2 successive frames we require

$$P_{2F} = P_{VM} P_{LM}$$

For 3 successive frames, we have the following options:

- A. Missing the first frame due to errors, but locking on the next two.
- B. Receiving all 3 frames successively, thus, Lock up occurred on first two.
- C. Missing the last frame due to errors, but Lock has occurred
(This is considered satisfactory since Lock occurred)

Thus:

$$P_{3F} = P_{VM} P_{LM} (\text{Probability of C}) + (\text{Probability of A}) P_{VM} P_{LM} \\ + P_{VM} P_{LM} (\text{Probability of B}) .$$

But

Probability of A = Probability we don't enter VM = $1 - P_{VM}$

Probability of B = Probability of zero errors for 3rd received
frame = P_{FC}

Probability of C = Probability of error in frame count = $1 - P_{FC}$

As a result,

$$P_{3F} = P_{VM} P_{LM} (1 - P_{FC}) + (1 - P_{VM}) P_{VM} P_{LM} + P_{VM} P_{LM} P_{FC} .$$

To enter VM from SM we must detect the frame sync pattern with zero errors and not have detected a false frame sync pattern (which would place us in an erroneous time slot, thus preventing the transition from VM to LM on the next frame). At the same time, there must be no errors in the frame count or the next frame count will not appear to have incremented (or decremented) properly. Thus,

$$P_{VM} = P_o (1 - P_{FS}) P_{FC} .$$

To enter LM from VM we must detect the next frame sync pattern with zero errors and not have an error in the frame count. (Note the possibility of an erroneous frame sync occurring in the data is not an option due to the time gating used in the detection algorithm.) Thus,

$$P_{LM} = P_o P_{FC} .$$

We must first determine the probability of an erroneous frame sync word, P_{FS} .

False sync can occur two ways:

- A. A slipped frame sync pattern can differ from the non slipped frame sync pattern by only a few bits. If these bits are changed by error on the channel, then false frame sync will occur.
- B. A false frame sync can occur in the data stream.

For situation A we note:

Frame Sync

1111 1010 1111 0011 0011 0100 0000 4 Bit ID

Slipped Frame Sync (two bits slip)

XX11 1110 1011 1100 1100 1101 0000 00XX

We note that since the 4 bit ID pattern is continuously changing it must be considered a random bit pattern and each bit may match the appropriate pattern with probability 1/2. Also note the slipped frame sync word matches the original in 14 places and differs in 12 places. To have frame sync occur within the slipped portion, we must thus have 12 bits changed by channel errors, 14 bits error free and two bits match by random choice. Thus, for this two bit slip we have the probability of false frame sync equal to

$$P_{FS}^{2 \text{ Bit Slip}} = (1/2)^2 (BER)^{12} (1 - BER)^{14}$$

or

$$P_{FS}^{2 \text{ Bit Slip}} \approx 2.4965 \times 10^{-61} (BER = 10^{-4})$$

A tabulation of the number of differences and number of random match digits versus the number of bits skipped is tabulated in Table I. Obviously a 28 bit slip produces a 28 random match situation and for a well chosen frame sync pattern no bit slips will produce agreement in all places. A worst case bound on the probability of false frame sync due to small slips in the bit stream may be

obtained by assuming each of the ± 27 bit slips produces 27 random matches and 1 difference. (BER = Channel Error Rate).

Thus,

$$P_{\text{FS}} \leq 54 (2)^{-27} (\text{BER}) \approx 4.0233 \cdot 10^{-12} \quad (\text{for } \text{BER} = 10^{-4}) \quad .$$

Bit Slips

In addition there are $3070 - 54 = 3016$ ways for random data matches to create false frame sync.

Thus

$$P_{\text{FS}} = 1.123547554 \cdot 10^{-5}.$$

Data Match

The total probability of false frame sync is thus bounded by

$$P_{\text{FS}} \leq 1.123547 \cdot 10^{-5} \quad (\text{for } \text{BER} = 10^{-4})$$

The probability of 0 errors in the frame count is

$$P_{\text{FC}} = (1 - \text{BER})^4 = (.9999)^4 \approx 9.9960006 \cdot 10^{-1} \quad (\text{for } \text{BER} = 10^{-4}) \quad .$$

Therefore, we have for P_{VM} (for $\text{BER} = 10^{-4}$)

$$P_{\text{VM}} \approx 9.967937 \cdot 10^{-1}$$

and for P_{LM} (for $\text{BER} = 10^{-4}$)

$$P_{\text{LM}} \approx 9.968049 \cdot 10^{-1} \quad .$$

For two successive frames, we have the probability of frame sync
Lock as: (BER = 10^{-4})

$$P_{2F} = P_{VM} P_{LM} \approx 9.936088 \cdot 10^{-1}$$

or

99.36088% of the time Lock up occurs within 2 frames.

For three successive frames we have the probability of frame sync
Lock as: (BER = 10^{-4})

$$P_{3F} = P_{VM} P_{LM} (2 - P_{VM}) \approx 9.9679465 \cdot 10^{-1}$$

or

99.67946% of the time Lock up occurs within 3 frames.

Table II shows the probability of frame sync Lock up for two and
three frames for BER's of 10^{-4} , 10^{-5} and 10^{-6} .

CONCLUSION

Dropping the 1 error tolerance in the 2MHZ frame sync detector
does not create an undesirable frame sync Lock up situation.

TABLE I

<u>NUMBER OF BIT SLIPS</u>	<u>NUMBER OF DIFFERENCES</u>	<u>NUMBER OF RANDOM MATCHES</u>
0	0	0
1	11	1
2	12	2
3	11	3
4	10	4
5	11	5
6	11	6
7	11	7
8	10	8
9	9	9
10	10	10
11	8	11
12	9	12
13	9	13
14	10	14
15	6	15
16	9	16
17	7	17
18	6	18
19	5	19
20	5	20
21	5	21
22	5	22
23	5	23
24	4	24
25	3	25
26	2	26
27	1	27
28	0	28

TABLE II
PROBABILITY OF FRAME SYNC LOCK UP VERSUS CHANNEL ERROR RATE (BER)

	BER = 10^{-4}	BER = 10^{-5}	BER = 10^{-6}
PERCENT OF TIME LOCK UP WITHIN 2 FRAMES WILL OCCUR	99.36088%	99.93489%	99.99567%
PERCENT OF TIME LOCK UP WITHIN 3 FRAMES WILL OCCUR	99.67946%	99.96799%	99.9968%

APPENDIX C

REPORT OF PDR (MARTIN/DENVER)

A report was submitted after attendance of the Preliminary Design Review (PDR) held at the Martin Marietta/Denver facility in March 1982. A copy of this report follows as Appendix C.

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

A Monthly Progress Report
Covering the Period
March 1, 1982 - May 31, 1982

Submitted to:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812
Technical Monitor: Mr. Russ Coffey, EF-13

Submitted by:

Mississippi State University
Engineering and Industrial Research Station
Department of Electrical Engineering
Mississippi State, MS 39762

Principal Investigator: Frank M. Ingels
Contract Number: NAS8-33887

A STUDY OF HIGH DENSITY BIT TRANSITION-
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

Work Summary (March 1, 1982 - May 31, 1982)

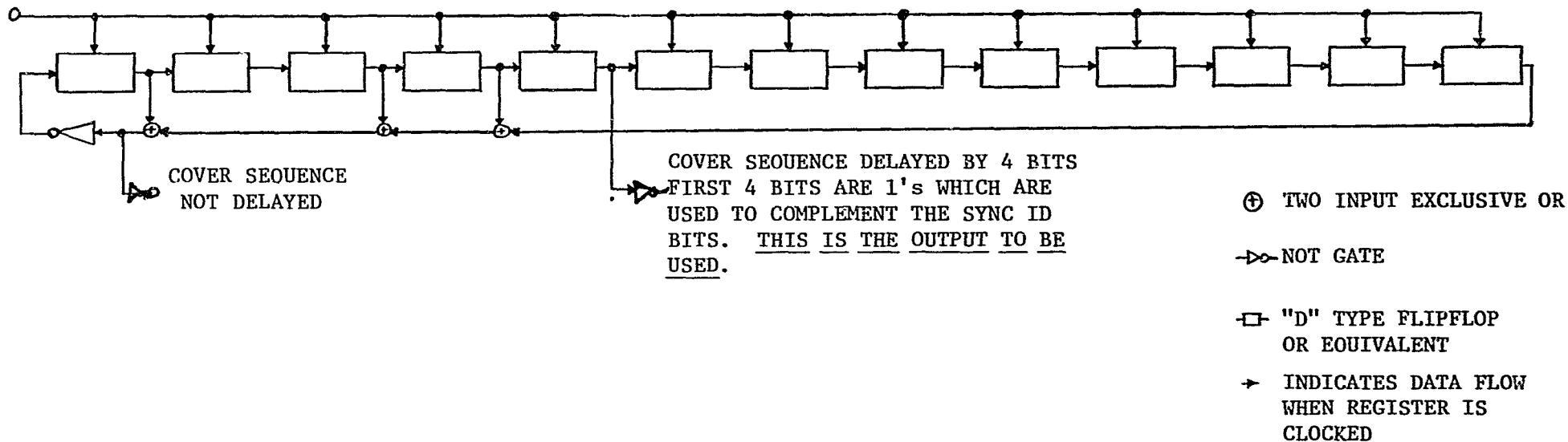
In March a preliminary design review (PDR) at Martin/Denver was attended. Participants included representatives and/or personnel from Martin/Denver, NASA/MSFC, NASA/KSC, M.B. & B./Germany, MATRA/France, MDTSCO/Huntsville.

A review of the actual encoder schematic to be implemented for the PN sequence generator was conducted by this participant. It was judged to be the same as the circuit recommended in the research report with two exceptions. These are; first that the circuit is now loaded with a clear line and the appropriate starting point is now achieved by use of complementing the appropriate outputs.

The CSG encoder decoder circuit approved is illustrated in Figure 1 of this report. Note, output off the fifth stage is required to achieve complementation of the sync ID bits.

Further questions will be clarified at the summer CDR meeting concerning the actual circuit (verification that it is as Figure 1) and to ascertain the operation of the counter used to start/stop the encoder.

SYNC PULSE
CLEAR TO "0"



REGISTER CONTENTS FIRST 15 SHIFTS

0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0	0	0	0	0	0
0	1	0	1	1	1	1	0	1	0	0	0	0	0
0	0	1	0	1	1	1	1	0	1	0	0	0	0
0	0	0	1	0	1	1	1	1	0	1	0	0	0
0	0	0	0	1	0	1	1	1	1	1	0	1	0
1	1	0	0	0	1	0	1	1	1	1	1	0	1
1	1	1	0	0	0	1	0	1	1	1	1	1	0
0	1	1	1	1	0	0	1	0	1	1	1	1	1

ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 1. CSG Encoder Register Circuit

APPENDIX D

REPORT OF CDR (MBB/MUNICH)

A report was submitted after attendance of the Critical Design Review (CDR) held at the MBB facility in Munich, Germany in June 1982. A copy of this report constitutes Appendix D.

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

A REPORT
COVERING THE
CRITICAL DESIGN REVIEW
(CDR)
June - July 1982

Submitted to:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812
Technical Monitor: Mr. Russ Coffey, EF-13

Submitted by:

Mississippi State University
Engineering and Industrial Research Station
Department of Electrical Engineering
Mississippi State, MS 39762

Principal Investigator: Frank M. Ingels
Contract Number: NAS8-33887

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

WORK SUMMARY (Critical Design Review - CDR)

During the period June 21, 1982 to June 26, 1982, a critical design review (CDR) for the cover sequence generator (CSG) encoder/decoder for Spacelab was held. This meeting was held at the Messerschmidt/Bohlm/Bolkow (MBB) facility in Ottobraun, Germany. Representatives from McDonnell Douglas Support Company (MTDSCO), Martin Marietta/Denver (MM/D), MATRA of France, ERNO and NASA/MSFC attended this meeting.

A demonstration of the engineering prototype unit test procedure was conducted. This demonstration encoded transmitted, received and decoded a known data stream. Various possible fault conditions were demonstrated along with the resulting front panel displays which indicated the fault under demonstration.

The demonstration tests were satisfactory in both performance and test design. Discussions with Dave Banerian of MM/D and Dick Burtzlauff (MM/D) to ascertain the test philosophy and test design procedure were conducted by this investigator. These discussions satisfied me as to the appropriateness of the testing procedure. Copies of the first annual report by Mississippi State University concerning the philosophy and design of the CSG encoder and decoder (in particular the sequence chosen and the properties desired) were requested by Mr. Jerry Malloy (MTDSCO), Mr. Don Rawson (MTDSCO) and Mr. Chuck Kervin (MBB). After clearing the request with Mr. Bern Siler of NASA/MSFC copies of these reports were distributed.

A discussion of the actual synchronization pattern being used in the Space Lab data format for the data being encoded/decoded arose when there was confusion over the nomenclature. The structure of the frame synchronization word is illustrated in Figure 1 of this report. Note the 28 bit synchronization pattern has the LSB of each 4 bit group on the left side while the 4 bit frame LD count is structured in an opposite fashion with the LSB on the right side. The proper hexadecimal code for the frame synchronization (28 bit) pattern is FAF3 340X where the X stands for the 4 bit ID frame count

which varies from frame to frame. The computer software used for the tests created the need to invert the data on a 4 bit basis.

There are six modes of HRDM operation. These are diagramed in Table 1. Displayed in Table 2 are the two operational modes of the HRM.

Martin Mrietta/Denver has verified all 3040 bits of the desired cover sequence pattern as generated by the HRM. Mr. Jerry Malloy and this investigator inspected the computer print out of the MBB demonstration test for approximately 300 bits and verified the synchronization word and proper incrementation of the ID count for seven consecutive synchronization patterns.

The test procedure of page 111 illustrates several routines used to verify that each synchronization bit is actually involved in the synch detector circuit operation. These routines developed by Martin/MBB are original and interesting.

It is worth mentioning that the analysis of the probability that frame synch lock is not acquired within 3 frames with and without a one error tolerance by this investigator and the analysis by Martin Marietta/Denver differ by less than .00019 percent. The difference lies in the inclusion by this investigator of the probability of error in the ID bits in the first frame which results in a discarding of that 3 frame sequence.

The CDR uncovered no problem, in design or philosophy. The CSG encoder/decoder is effectively a finished product from the philosophy and design viewpoints.

NORMAL STRUCTURE	1111 1010 1111 0011 0011 0100 0000 XXXX
STRUCTURE INVERTED ON LETTER BY LETTER BASIS	<u>1111 0101 1111 1100 1100 0010 0000 XXXX</u> L M L M M L S S S S S S B B B B B B

Figure 1. Structure of the Frame Synchronization Word

TABLE I.

OPERATIONAL MODES OF HRDM

<u>OPERATIONAL MODE</u>	<u>NORMAL</u>	<u>ENCODER FAILURE</u>	<u>DECODER FAILURE</u>	<u>BOTH FAIL</u>
1. Normal Unencoded	L	L	L	L
2. Inverted-Unencoded	L	L	L	L
3. Reversed-Unencoded	L	L	L	L
4. Reversed-Inverted	L	L	L	L
5. Normal-Encoded	L+E	L	(F) L+E	L
6. Inverted-Encoded	L+E	L	(F) L+E	L
7. Decoder-Self Test	L	N/A	E	N/A

FRONT PANEL HRDM LIGHTS:

L = Lock Indicator Light On

E = Encoder On Light Lit

F = Decoder Fail Signal (Output on Back Panel of HRDM).

(This occurs when either a hard or soft error occurs in the decoder but the encoder is on this situation continues until frame lock is lost. Then decoder may retry to work.)

TABLE 2

OPERATIONAL MODES OF HRM

<u>OPERATIONAL MODE</u>	<u>NORMAL</u>	<u>FAIL</u>	
NORMAL	CSG ON + GO	CSG OFF + NO GO	NO GO = FAIL*
BITE (All Outputs Unencoded)	CSG OFF + GO	CSG OFF + NO GO	NO GO = FAIL*

* Single failure saved until saved until Housekeeping readout.

- a) One time error readout = soft failure
- b) Permanent error readout = hard failure

(See pages 1-5 of MBB's CDR package)

APPENDIX E

INTERIM FINAL REPORT

JULY 1, 1980 to DECEMBER 31, 1981

This appendix contains the interim final report which fully documents the technical design, philosophy and system constraints behind the development of the CSG circuitry. This system has been built, tested and incorporated into the HRM and HRDM units by MBB/Munich and Martin Marietta/Denver under the auspices of MDTSCO/Huntsville.

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

Interim
Final Report
Covering the Period
July 1, 1980 - December 31, 1981

Submitted to:

Mr. Russ Coffee EF-13
George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama
35812

Submitted by:

Mississippi State University
Engineering and Industrial Research Station
Department of Electrical Engineering
Mississippi State, Mississippi
39762

Principal Investigator: Frank Ingels
Associate Investigator: William O. Schoggen

Contract No. NAS8-33887

EXECUTIVE SUMMARY

Digital data streams using non-return-to-zero-level encoding (NRZ-L) as the signalling waveform are subject to periods of no level change. The signalling waveform with not changing levels has energy content that is rich about the origin and of course does not contain any edges in the waveform.

Proper operation of the ground station receiving such digital data streams depends upon satisfactory bit synchronization. Bit synchronizers typically require a certain minimum percentage of edges in the received signalling waveform. This requirement is not always satisfied by NRZ-L waveforms.

The purpose of this investigative study has been to determine a satisfactory method of providing sufficient bit transitions (edges) in the signalling waveform for the 2 MHz data link of the Space Shuttle High Rate Multiplexer (HRM) unit.

The system design already in existence places several constraints on any method used to ensure the desired bit transition density of at least 1 bit transition in every 64 bits and at least 64 bit transitions in every 512 bits.

These constraints are:

1. The method chosen must produce at least 1 bit transition in every 64 bits and at least 64 transitions in every 512 bits.
2. The method chosen must not increase the present bandwidth nor decrease the present information rate.
3. The method chosen must be compatible with the existing BCH code.
4. The method chosen should have a minimal design impact on the present system.
5. The method chosen must pass unaltered any data stream whose data rate is greater than 2 MHz.
6. The method chosen should resolve the bit phase ambiguity problem inherent in the Channel 2 return link of the KU-Band system.

Many methods for increasing bit transition densities in a data stream exist. These methods have been summarized, discussed in detail and compared one against another and against the constraints mentioned above.

These methods include use of alternate Pulse Code Modulation (PCM) waveforms, data stream modification by insertion, alternate bit inversion, differential encoding, error encoding and use of bit scramblers. (Bit scramblers come in many different versions such as: self synchronizing, multi and single count, serial, cascaded and parallel scramblers).

Of all the methods discussed, one method satisfied the desired objective, met all constraints and had advantages that outweighed disadvantages when compared against the remaining methods. This method was chosen - the reset scrambler or simply the Pseudo-Random Cover Sequence Generator (PN-CSG). This technique is fully analyzed and a design implementation is proposed.

The method consists of modulo-2 addition of a PN sequence to the data stream before the modulations of the Radio Frequency (RF) transmitter. It is recommended that only the data streams and the 4 bit frame synchronization Identity Count (ID) be so modified. It is recommended that there be no change in the 28 bit frame synchronization word.

If the PN sequence is added to the frame synchronization word, it is very likely that the special properties chosen for frame synchronization patterns would be violated. Furthermore, the decoder must then search for four frame synchronization patterns if it is desired to correctly detect the presence or absence of the PN cover sequence, and to alleviate any phase ambiguity.

The probability of failing to provide the required bit transition density is less than $2.44 \cdot 10^{-17}$. A computer simulation program was developed which tests the truncated PN sequence with random data streams. The computer results indicate that the output sequence to be transmitted by the RF modulator will have a transition density of approximately 50%. This should improve the overall KU Band system performance considerably in the presence of low signal-to-noise ratios by increasing the bit synchronizer's capability to stay locked to the incoming bit clock.

The statement of work contains five distinct items in five distinct paragraphs. The first three items are specifically addressed in sections 2 and 3 with Table 3.1 presenting a summary of the various methods of HBTD encoding and comparing their relative performance in so far as error propagation characteristics, transition properties and system constraints are concerned. The appendix contains a computer simulation of the system using the specific PN code recommended in this study. The interim report of September 30, 1980 recommended a specific PN sequence and this is detailed in section 4.B.

TABLE OF CONTENTS

Chapter		Page
	EXECUTIVE SUMMARY	i
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
	LIST OF SYMBOLS	viii
1	INTRODUCTION	1
2	STATEMENT OF PROBLEM AND CONSTRAINTS	7
	2.A THE PROBLEM	7
	2.B PRIMARY SYSTEM CONSTRAINTS	7
	2.C SECONDARY SYSTEM CONSTRAINTS	11
3	DIFFERENT METHODS AVAILABLE TO ALLEVIATE THE PROBLEM	13
	3.A ALTERNATE PULSE CODE MODULATION WAVEFORMS	13
	3.A.1 B1-PHASE	13
	3.A.2 DELAY MODULATION	13
	3.B DATA STREAM MODIFICATION	16
	3.B.1 ALTERNATE BIT INVERSION	17
	3.B.2 DIFFERENTIAL ENCODING	17
	3.B.3 BIT INSERTION	18
	3.B.4 ERROR ENCODING	19
	3.B.5 BIT SCRAMBLERS	25
	3.B.5.a SELF-SYNCHRONIZING SCRAMBLERS	25
	3.B.5.a.1 MULTI-COUNTER SCRAMBLER	29
	3.B.5.a.2 SINGLE-COUNTER SCRAMBLER	30
	3.B.5.a.3 TRANSITION DENSITY FOR SELF-SYNCHRONIZING DESCRAMBLERS	30
	3.B.5.a.4 SELF-SYNCHRONIZING DESCRAMBLERS	31
	3.B.5.a.5 THE SPECTRUM OF THE SCRAMBLER OUTPUT	34
	3.B.5.a.6 SERIAL, CASCADED, AND PARALLEL SCRAMBLER	35
	3.B.5.b NON-SELF-SYNCHRONIZING (RESET) SCRAMBLERS	38

TABLE OF CONTENTS (Continued)

Chapter		Page
4	PSEUDO-NOISE COVER SEQUENCE (RESET SCRAMBLER)	41
	4.A PSEUDO-NOISE (PN) SEQUENCE	41
	4.B THE PARTICULAR PN SEQUENCE FOR THE COVER SEQUENCE FOR THE COVER SEQUENCEB GENERATOR (CSG) .	44
	4.C PROBABILITIES ASSOCIATED WITH THE SEQUENCE . . .	47
5	CONCLUSIONS AND RECOMMENDATIONS	54
6	REFERENCES	57
7	APPENDICES	59
	7.A COMPUTER SIMULATION PROGRAM	60
	7.B SPECIFICATION DOCUMENT FOR PN CSG ENCODER/DECODER .	76
	7.C NOVEMBER 1980 MONTHLY REPORT	86

LIST OF FIGURES

Figure		Page
1.1	KU Band Return Link	2
1.2	HRM - KUSP Interface	4
1.3	General User's Format	6
3.1	PCM Waveforms	14
3.2	1/3 Convolutional Encoder with Alternated Bit Inversion Employed by Space Telescope	21
3.3	Periodic Convolutional Interleaver and Deinterleaver Employed by Space Telescope.	24
3.4	Basic Self-Sync Scrambler	26
3.5	Multi-Counter Scrambler	27
3.6	Single-Counter Scrambler	28
3.7	Multi-Counter Descrambler	32
3.8	Single-Counter Descrambler.	33
3.9	Cascading of N M-bit Scramblers	36
3.10	Parallel Scrambler Configuration.	37
3.11	Reset Scrambler and Descrambler (PN Cover Sequence Generator)	39
4.1	General PN Cover Sequence Generator.	42
4.2	CSG Encoder Using Shift Register Implementation.	45
4.3	CSG Encoder Using ROM Implementation	46
B.1a	CSG Encoder Functional Location	79
B.1b	CSG Decoder Functional Location	79
B.2	HRM Data Stream	80
B.3	Basic Cover Sequence Encoder Block Diagram	82
B.4	Cover Sequence Decoder Block Diagram	83
C.1	Modified Cover Sequence Encoder Block Diagram	89
C.2	Modified Cover Sequence Decoder Block Diagram	90

LIST OF TABLES

Table		Page
1.1	Available Input Data via KU-Band Link	2
2.1	Main System Constraints	8
2.2	Secondary System Constraints	12
3.1	Bit Transition Density Coding Choices	40
4.1	Zero-One Distribution for n=13 Maximal Length PN Sequence	48
4.2	Truncated Sequence for the CSG	49
4.3	Zero-One Distribution for the Truncated Sequence	51
4.4	Alternate One-Zero Distribution for the Truncated Sequence	52
5.1	Comparison of CSG With System Constraints	56

LIST OF SYMBOLS

BCH	Base-Chanduri-Hocquenghem
BER	Bit-Error-Rate
BSR	Bit-Slip-Rate
BSS	Basic Self-Sync Scrambler
CSG	Cover Sequence Generator
DM	Delay Modulation
FBSRG	Feedback Shift Register Generator
FM	Frequency Modulation
HBTD	High Bit Transition Density
HRAA	High Rate Acquisition Assembly
HRDM	High Rate Demultiplexer
HRM	High Rate Multiplexer
KUSP	Space Shuttle KU-Band Signal Processor
LCM	Least Common Multiple
Mbps	Mega Bits Per Second
MCS	Multi-Counter Scrambler
ML	Maximal Length
PCM	Pulse Code Modulation
QPSK	Quadrature Phase Shift Keying
SCS	Single-Counter Scrambler
SL	Spacelab
SR	Shift Register
ST	Space Telescope
TDRSS	Tracking and Delay Relay Satellite System

CHAPTER 1

INTRODUCTION

The Spacelab (SL) is an orbital laboratory which remains attached to the Space Shuttle for the duration of a mission. The SL will utilize the KU-Band communication system of the Shuttle for communication with the ground. Figure 1.1 is a simplified sketch of the return link. The following paragraphs give a brief description of the SL return link, for further information the reader is referred to References 1, 2, 3, and 4.

The onboard experiment data is collected by the High Rate Acquisition Assembly (HRAA). The HRAA consists of the onboard High Rate Multiplexer (HRM) plus associated High Rate Links, and the ground based High Rate Demultiplexer (HRDM). The return link, connecting the HRM and HRDM, utilizes the Shuttle KU-Band Signal Processor (KUSP) and the Tracking and Delay Relay Satellite System (TDRSS) which includes the receiving station, and the bit synchronizer. The return link will provide a bit-error-rate (BER) of at least 1×10^{-5} with a Bit-Slip-Rate (BSR) of less than 1×10^{-15} provided certain system constraints are met. Herein lies the problem; the SL data stream violates the transition requirements of the bit synchronizer. This will be further reviewed in Chapter II.

The HRM receives data from 18 experiments, 2 I/O units and 2 records, and outputs data to the KUSP and the 2 recorders. Sixteen of the 18 experiment channels are switchable and the other 2 are direct. The HRM time multiplexes these 16 channels with the other data including playback data from the recorders and serially transmits the data to the KUSP.

The SL employs the KU-Band Link of the Shuttle and this link utilizes the two operation modes of the KUSP; Mode 1 - Quadrature Phase Shift Key (QPSK) modulation and Mode 2-FM modulation. The KUSP has three channels of input data in each mode. Table 1.1 lists the available input data in both modes. Figure 1.2 illustrates the interface between the HRM and the KUSP.

The HRM uses several different format structures for its output. Only the general user format, for frequencies less than 32 Mbps, will

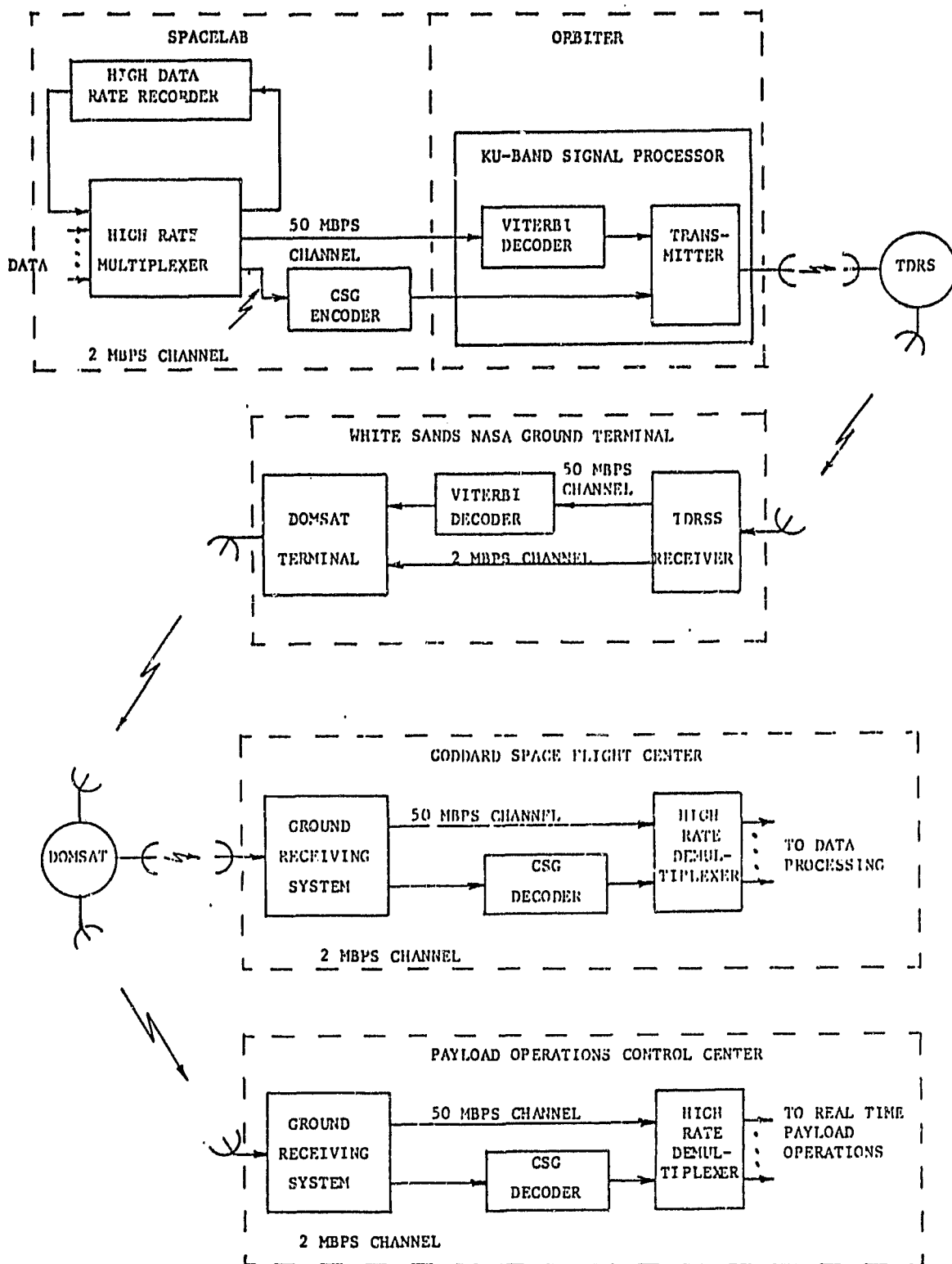


Figure 1.1 KU Band Return Link

TABLE 1.1 AVAILABLE INPUT DATA VIA
KU-BAND LINK

FM MODE 1:

CHANNEL 1 - SAME AS FM Mode

CHANNEL 2 - SAME AS FM MODE

CHANNEL 3 - PAYLOAD DIGITAL OUTPUT, 2-50 MBPS WITH CLOCK,
NRZ-L, M, OR S

FM MODE 2:

CHANNEL 1 - 192 KBPS B1-PHASE-L PCM AND VOICE FROM NSP

CHANNEL 2 - SELECT ONE OF FOUR INPUTS

- PLI (NARROW BAND BENT PIPE FROM DETEACHED PAYLOAD)
- PAYLOAD DIGITAL OUTPUT, 16 KBPS-2MEPS, NRZ-L, M,
OR S, OR 16 KBPS - 1.024 MBPS B1-PHASE L, M, OR S
- OPERATIONAL RECORDER DUMP
- PAYLOAD RECORDER DUMP

CHANNEL 3 - SELECT ONE OF FOUR INPUTS

- TV
- PAYLOAD DIGITAL OUTPUT, 16 KBPS-4 MBPS, NRZ-L, M, OR S
- PAYLOAD ANALOG OUTPUT, DC-4.5 MHz
- PLI (WIDE BAND BENT PIPE FROM DETACHED PAYLOAD)

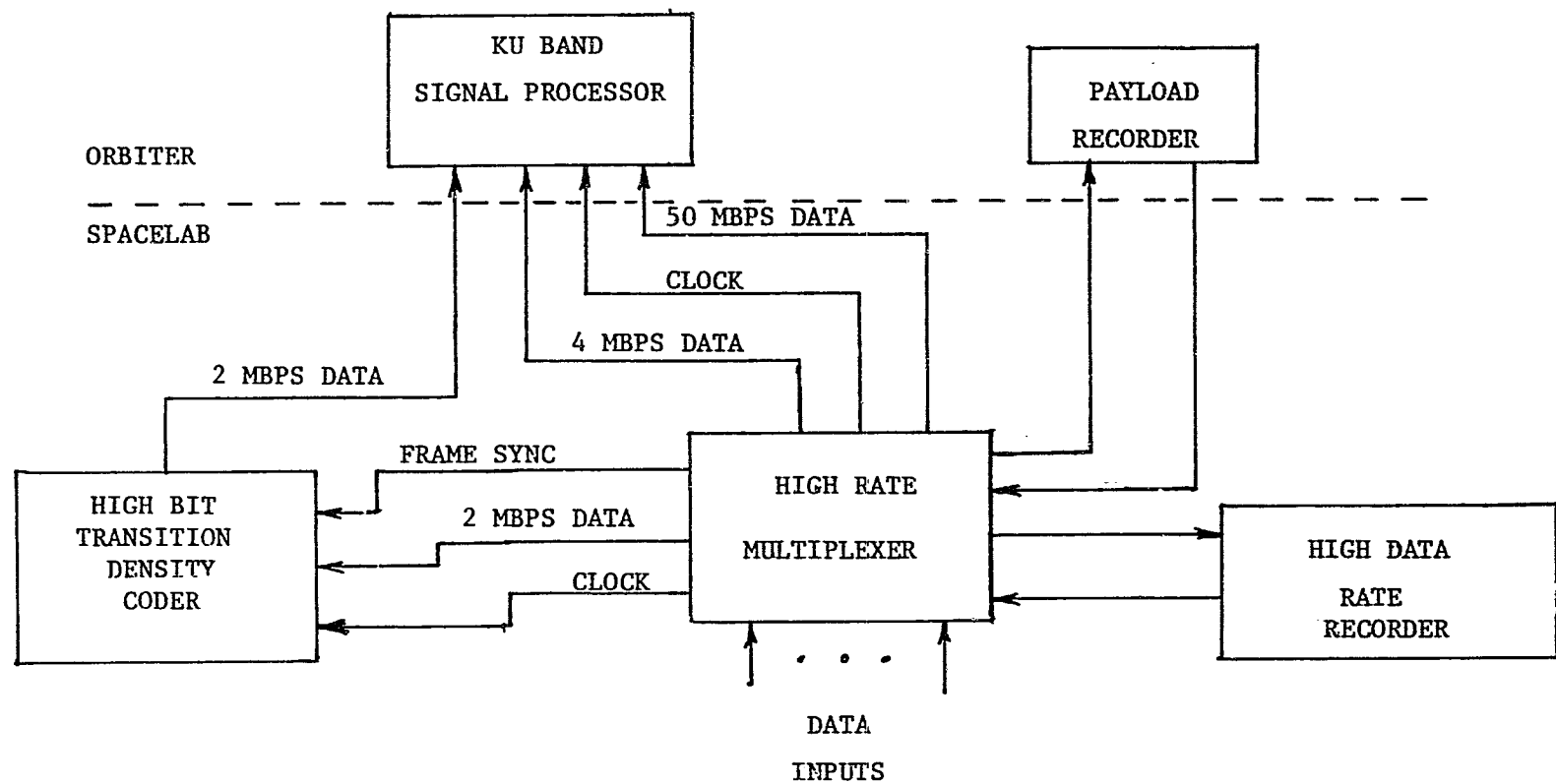


Figure 1.2 High Rate Multiplexer to KU-Band Signal Processor Interface

be discussed. For further information concerning the HRM formats; see references 1 and 2. The user format consists of eight frames of 96 words each (768 words), each frame beginning with a sync or a status word. The normal frame is composed of 6 lines by 16 words. Since each of the 21 inputs to the HRM operate at bit rates asynchronous to the output bit rate, fill data is required. The 16th or last word of each line is a fill data indicator. The HRM employs a unique method to provide the necessary fill data words. When a line of data requires fill, all the valid data words to the right of the fill word are shifted to the left thus making the fill word the 15th word of the data line. The system requires that the fill data (stuffing) indicator shall be constructed such that the probability of error in interpreting the indicator is less than 10^{-10} . The HRM utilizes a (31,16,3) BCH code to satisfy the required BER for the stuffing indicator. The HRM encodes the 16 bit stuffing indicator into a 31 bit BCH code word capable of correcting 3 errors per word. When more than one fill word is needed, the valid data words are again shifted left and the 14th word of the previous line is inserted into the 14th word of the present line.

The general input data format is illustrated in Figure 1.3. The input data to the HRM is as follows:

18 experiments	NRZ-L	
HDRR	NRZ-L	Reverse playback 2 Mbp only
PLR	Manchester II	Reverse 1Mbp only
		(Must be playback through HRDM)

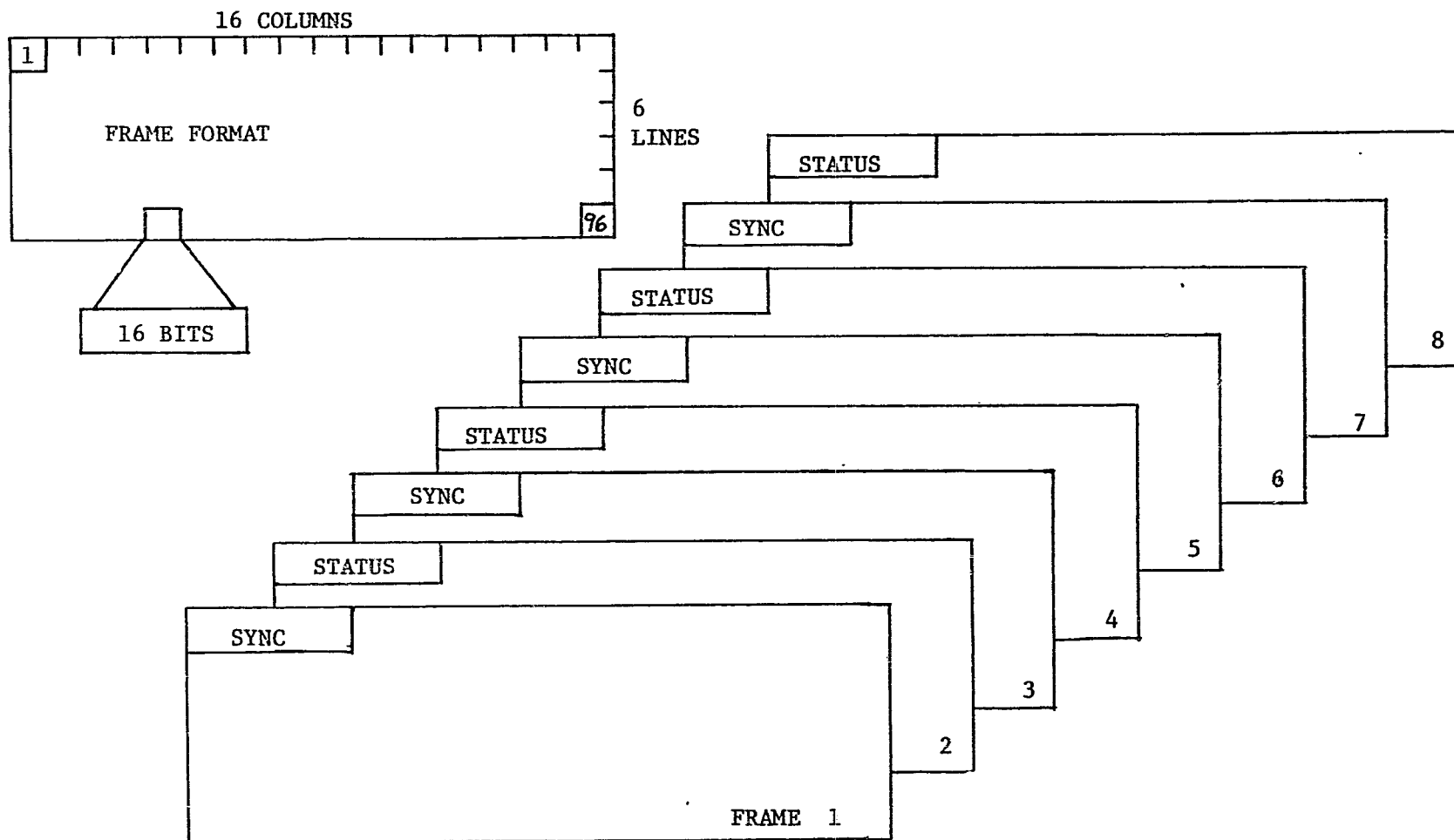


Figure 1.3 General User Format

CHAPTER 2

STATEMENT OF PROBLEM AND CONSTRAINTS

2.A THE PROBLEM

An investigation of the effect of low bit transition density on the performance of the Channel 2 KU-Band return link was conducted. The results of this investigation indicated the SL will not meet the minimum specifications of the TDRSS Users' Guide with respect to the bit synchronizer. The end result of this failure to meet the bit synchronizer (sync) specifications is loss of lock by the bit sync and subsequent loss of data for an undefined period of time (Reference 5). To alleviate this problem the SL must increase the bit transition density on the Channel 2 KU-Band return link to a minimum of one bit transition every sixty-four bits and sixty-four bit transitions within five-hundred-twelve bits.

There are several different methods that may be employed to provide the data stream with the necessary bit transition density. These methods are discussed in more detail in Chapter III. This chapter deals mainly with the problem and the constraints caused by the system. There are four main constraints placed on the HBTD coder. They are listed in Table 2.1 and discussed in the following paragraphs.

2.B PRIMARY SYSTEM CONSTRAINTS

The primary objective of the HBTD modification is to increase the bit transition density of the data stream to at least one bit transition every sixty-four channel symbols and sixty-four bit transitions every five-hundred-twelve channel symbols. This is the absolute minimum requirement by the TDRSS bit synchronizer at White Sands, New Mexico to provide the desired BSR of 10^{-15} with the SL signal characteristics.

Since the 2Mbit channel is expected to operate at the maximum rate of 2 Mbps for maximum utilization, the modification must not increase the channel errors. The system requires that the fill indicator must be able to locate the fill data with an accuracy of no less than 10^{-10} . The fill indicator is a (31,16,3) BCH code. A (31,16,3) BCH code takes 16 information bits and converts them into a 31 bit code word capable of correcting 3 errors. Since the channel bit error rate affects the fill

1. THE HBTD CODE MUST HAVE AT LEAST ONE TRANSITION EVERY 64 BITS
AND AT LEAST 64 TRANSITIONS WITHIN 512 BITS
2. THE CODE MUST NOT INCREASE THE PRESENT BANDWIDTH, NOR DECREASE
THE INFORMATION RATE.
3. IT MUST BE COMPATIBLE WITH THE EXISTING BCH CODE. (IMPLIES A SCHEME
WHICH PRODUCES MULTIPLE DECODER ERRORS PER DISCRETE CHANNEL ERRORS
WOULD BE INCOMPATIBLE WITH BCH CODE.)
4. THE HARDWARE IMPLEMENTATION MUST HAVE MINIMAL DESIGN IMPACT ON THE
PRESENT SYSTEM.

indicator word, any modification that increases the channel error would degrade the fill indicator. For example, if the system modification produced only one additional error per channel error, then the reliability of the fill indicator, as shown below, would be 4.6×10^{-8} , which is unacceptable.

To illustrate the problem consider the effects of differential encoding on BCH word error probability. Assuming the RF channel has an average random error rate, due to additive white gaussian noise, corresponding to 1 bit in 100,000 (that is an average bit error rate of 1×10^{-5}) one may calculate the probability of an erroneous decoding of a BCH coded word with use of NRZ-L code and also with use of a differential encoding such as NRZ-M or NRZ-S.

The model to be used is:

Using NRZ-L Coded Data Stream

A single error on the RF channel in the BCH code word will result in a single error in the data input to the BCH decoder. The BCH code word can correct up to and including 3 errors out of 31 bits.

Thus the probability of erroneous decoding of the BCH word (PEBCH) is

$$\text{PEBCH} = 1 - [P(0) + P(1) + P(2) + P(3)] \quad (2.1)$$

where:

$P(0)$ = Probability of no errors in the 31 bit word

$P(1)$ = Probability of 1 error in the 31 bit word

$P(2)$ = Probability of 2 errors in the 31 bit word

$P(3)$ = Probability of 3 errors in the 31 bit word

In general $P(X)$ is expressed as

$$P(X) = \binom{N}{X} p^X q^{N-X} \quad (2.2)$$

where

p is the probability of an RF channel error

q is the $1 - p$

X is the number of errors in the word

N is the number of bits in the word.

The expression $\binom{N}{X}$ relates to the number of different ways in which X errors occur in an N bit word and

$$\binom{N}{X} = \frac{N!}{(N-X)!X!} \quad (2.3)$$

Using NRZ-M or NRZ-S (Differential Encoding) Coded Data Stream

A single error on the RF channel in the BCH code word will result in two adjacent errors on the data stream input to the BCH decoder. (Reference is any text in communications, in particular: Reference 16, page 324.)

Thus we have as the probability of erroneous decoding of the BCH word with Differential Encoding (PEBCHDE)

$$\text{PEBCHDE} = 1 - [P(0) + P(1)] \quad (2.4)$$

Where the expressions $P(0)$ and $P(1)$ are as defined in the previous case. Note this expression for PEBCHDE reflects the fact that 2 RF channel errors in the BCH word will result in 4 errors presented to the BCH decoder and this will result in erroneous decoding.

These calculations were performed using a double precision digital computer program.

The results are

$$\begin{aligned} \text{PEBCH} &= 3.01841884819964434 \times 10^{-16} \\ \text{PEBCHDE} &= 4.64910190316402087 \times 10^{-8} \end{aligned}$$

One sees a significant difference in the error performance due to erroneous decoding of BCH words!

For a 50 Mbps data stream the average length of time between erroneous decoding of the BCH word would be approximately

160 seconds average between erroneous decoding of BCH words
using differential encoding (NRZ-M or NRZ-S)

16,000,000,000 seconds average between erroneous decoding of
BCH words using NRZ-L encoding.

The unacceptability of adding additional errors to the system is obvious.

The fourth criterion, minimal impact on the present system, results from the fact that the system is in the production stage and any major changes would be very costly. Approximately \$100,000 cost results from a minor change alone due to the paper work required.

2.C SECONDARY SYSTEM CONSTRAINTS

Additional criteria, dealing with the implementation, results from the expected characteristics of the data stream. The modification must pass unaltered data emanating from any source other than the HRM and data rates greater than 2 Mbps from the HRM. These constraints result from the physical location of the modifications and the operational functions of the HRM. All data emitted from the HRM via the 2 Mbit channel will be NRZ-L. The above stated constraints apply to both the encoder and decoder. The decoder must also resolve the phase ambiguity problem which results when a Bi-Phase NRZ-L data is used. Since NRZ-L employs a high level to represent a one and low level to represent a zero, it is possible for the data stream to become inverted. That is to say a transmitted one is received as a zero and vice versa. Therefore the decoder must be capable of detecting and correcting the inverted data stream. These secondary system constraints are listed in Table 2.2.

TABLE 2.2 SECONDARY SYSTEM CONSTRAINTS

1. HBTD CODE MUST PASS UNALTER ANY DATA STREAM WHOSE RATE IS
GREATER THAN 2 MBPS.
2. THE CODE MUST PASS UNALTER ALL DATA STREAMS WHICH EMANATE
FROM SOURCES OTHER THAN THE HRM REGARDLESS OF THE DATA RATE.
3. THE HBTD ENCODER MUST HAVE A BYPASS MODE.
4. THE HBTD CODE MUST RESOLVE THE "BIT AMBIGUITY"
PROBLEM INHERENT TO THE CHANNEL 2 RETURN LINK.

CHAPTER 3

DIFFERENT METHODS AVAILABLE TO ALLEVIATE THE PROBLEM

Two basic types of modifications exist to improve the Bit Transition Density of the Channel 2 data stream to the minimum requirements of the bit synchronizer. They are: (1) Use an alternative PCM waveform or (2) modify the data stream. The remainder of the section is devoted to describing several methods for accomplishing these modifications.

3.A ALTERNATE PCM WAVEFORM

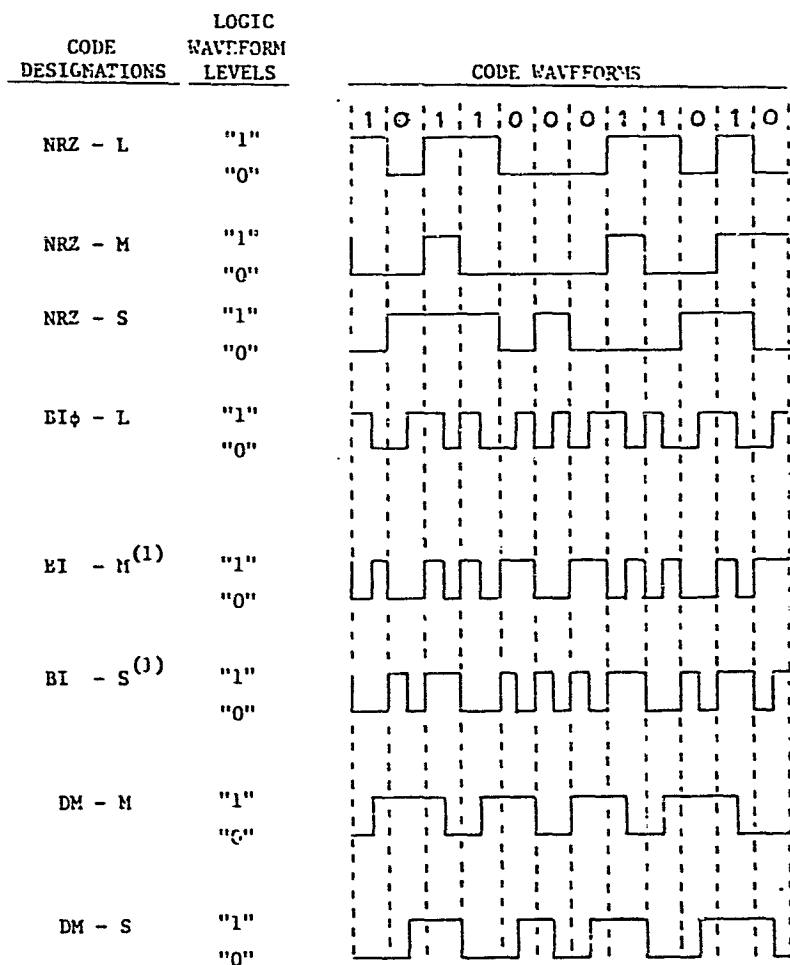
The 2 Mbit channel presently employs a NRZ-L waveform. There are several other binary waveforms available. Several of the most common are shown in Figure 3.1. The most frequently used waveforms, for high bit transition density applications, are the Bi-Phase and Delay Modulation waveforms. Both are widely used in the tape recording industry.

3.A.1 Bi-Phase

The three main types of Bi-Phase waveforms are; Bi-Phase Level, Bi-Phase-Mark, and Bi-Phase-Space. Bi-Phase Level is also called Split Phase or Manchester Code. All three waveforms provide at least one transition for each bit cell. They produce single output errors for a single input or channel error. The hardware required to implement each is moderate in complexity. All three Bi-Phase waveforms are self-synchronizing. The main disadvantage of all three is that Bi-Phase modulation requires twice the bandwidth of the present NRZ-L to provide the same information rate. Therefore the use of Bi-Phase would require either an increase in the present bandwidth or a decrease in the information rate. Neither case is acceptable since it violates one of the main system constraints.

3.A.2 Delay Modulation

Delay Modulation (DM) is a procedure for encoding binary data into rectangular waveforms of two levels according to the following rules for DM-M:



CODE DEFINITIONS

NON-RETURN-TO-ZERO-LEVEL

"One" is represented by one level.

"Zero" is represented by the other level.

NON-RETURN-TO-ZERO-MARK

"One" is represented by a change in level.

"Zero" is represented by no change in level.

NON-RETURN-TO-ZERO-SPACE

"One" is represented by no change in level.

"Zero" is represented by a change in level.

BI-PHASE LEVEL (SPLIT PHASE)

Level change occurs at center of every bit period.

"One" is represented by a "one" level with the transition to the "zero" level.

"Zero" is represented by a "zero" level with the transition to the "one" level.

BI-PHASE-MARK

Level change occurs at the beginning of every bit period.

"One" is represented by a midbit level change.

"Zero" is represented by no midbit level change.

BI-PHASE-SPACE

Level change occurs at the beginning of every bit period.

"One" is represented by no midbit level change.

"Zero" is represented by a midbit level change.

DELAY MODULATION-MARK (MILLER CODE)

"One" is represented by a level change midbit time.

"Zero" followed by a "zero" is represented by a level change at the end of the first "Zero" bit. No level change occurs when a "zero" is preceded by a "one".

DELAY MODULATION-SPACE (MILLER CODE)

"Zero" is represented by a level change at midbit time.

"One" followed by a "one" is represented by a transition at the end of the first "one" bit. No level change occurs when a "one" is preceded by a "zero".

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3.1 PCM Waveforms

1. A one is represented by a transition from one level to the other at the midpoint of the bit cell.
2. A zero is represented by no transition unless it is followed by another zero. The case of consecutive zeros is represented by a transition at the end of the leading zero bit cell.

In the case of DM-S the rules for ones and zero are enterchanged. These rules are illustrated in Figure 3.1.

Delay Modulation has several attractive properties:

1. The majority of the signalling energy lies in frequencies less than one-half the symbol rate.
2. The power spectrum is small in the vicinity of $f = 0$ (that is at D.C.).
3. DM provides at most one transition per bit cell and at the least 2 bit transitions every 3 bit cells; thus, providing a bit stream with a very high bit transition density.

These properties provide DM with the advantage of inherent self-timing information using phase modulation which is not present in NRZ-L, while requiring approximately the same bandwidth as NRZ-L. DM is also suitable for use with tape recorders, especially when higher packing density is required, or with systems which require high bit transition densities.

DM requires a given 3 bit sequence to assure proper bit sync. This sequence is 101 for DM-M. This sequence has a high probability of occurring one or more times in any random data bit stream. The probability that one or more 101 bit sequences will occur increases rapidly as the number of bits in the data sequence increases. The following equation may be used to obtain a close approximation of the probability of 101 occurring n or more times in m bits (the number of bits per sequence).

$$P_m(101 \geq n) = 1 - P_m(101 \leq n) = 1 - (P_m(101=0) + P_m(101=1) + \dots$$

$$P_m(101=(n-1)) \tag{3.1}$$

Where:

$$P_m(101=r) = \binom{k}{r} \left(q_0^{k-r} \right) \left(p_0^r \right)$$

q_0 = the probability of any 3 bits not being 101 = $\frac{7}{8}$

p_0 = the probability of any 3 bits being 101 = $\frac{1}{8}$

$$k = m - 2$$

$$\binom{k}{r} = \frac{k!}{(k-r)!r!}$$

For example, let m be 16 (for 14 binary bits) then the probability of a 101-bit sequence occurring one or more times is:

$$P_{16}(101=1) = 1 - P(101=0)$$

$$= 1 - \binom{14}{0} \left(\frac{7}{8} \right)^{14} \left(\frac{1}{8} \right)^0 = 1 - \frac{14!}{0!14!} \left(\frac{7}{8} \right)^{14} = 1 - \left(\frac{7}{8} \right)^{14} = 1 - .154 = .846$$

(3.2)

In other words, there is a 84.6% probability of a 101 pattern occurring and hence providing bit sync within a 16 bit sequence. Thus, one should expect a bit sync lock within a very short time upon the start up of a DM encoded sequence. The main disadvantage of Delay Modulation is that single errors into the decoder will yield double errors out of the decoder. This results from the comparison of the present and most recent bit to determine the value of the previous bit. Thus the property that produces the improved bit transition density also makes DM incompatible with the BCH code used by the fill indicator.

3.B DATA STREAM MODIFICATION

There are several means of modifying the present data stream to meet the bit transition requirements. The first method that comes to mind is to simply invert every other bit or alternate bit inversion. Other means such as differential encoding, a bit insertion technique or error correcting encoding techniques are commonly used to improve

the bit transition density of a data stream. Bit scrambling is another technique which is employed to increase the bit transition in a sequence. Each of these methods are discussed in the following paragraphs.

All the aforementioned techniques require the same bandwidth for the same information rate as the present system.

3.B.1 Alternate Bit Inversion

Alternate bit inversion is probably the simplest method for increasing the transition density. This technique inverts every other bit of the data stream. It yields excellent results in the case of long sequences of bits of the same value. The implementation is very simple except for the synchronization with the data stream and a bit slip will result in the inversion of the original data stream. Single channel errors produce single output errors. The main disadvantage of this technique is the inability to guarantee the bit transition density. In the case of an alternating input data stream, this procedure will produce an output sequence of bits of the same value equal in length to the input sequence. Since the SL data stream is expected to contain long runs of alternating bits this method must be discarded. Additional logic could be added to the encoder and decoder to prevent this occurrence but the logic required would be quite complex and require additional considerations.

3.B.2 Differential Encoding

The use of differential encoding as a means of improving the bit transitions density has received considerable use in other systems, especially when the data stream contains long strings of bits having the same value; all ones or all zeros. This technique has advantage over alternate bit inversion in that an alternating bit sequence retains half of the original transitions. There are two types of differential coding, NRZ-M and NRZ-S as illustrated in Figure 3.1. NRZ-M uses a change of state to indicate a one and no change for zero, while NRZ-S indicates a zero by a change in level and no change for a one. Differential encoding provides a 50% transition density for an alternating input data stream and a 100% transition density for sequence of the same value provided the value is the one represented

by a change of state, one for NRZ-M and zero for NRZ-S. If the values is the level represented by no change, nothing is gained by different encoding. Therefore one must design for the level which is most dominant in the data stream and know that the sequence lengths of the other level will not result in a loss of lock by the bit sync. Another disadvantage of differential encoding is that single input errors produce double output errors. Since the differential encoding propagates the number of channel errors and can be designed for only a single case of bit sequence of the same value instead of for both, it must be eliminated as a possible solution.

3.B.3 Bit Insertion

Bit insertion technique are also commonly used to increase bit transition density. There are several different types of bit insertion techniques. Some add bits to the data stream while others use blocks of bits to replace certain data sequences. However, all of these techniques share the need for a complex timing and counting circuitry. The basic concept for all insertion techniques is the need to recognize when to insert and when to remove their specified patterns. For example, assume a bit replacement technique is to be used to meet the sixty-four in five-hundred-twelve requirement. The obvious way to guarantee that the data stream would meet this requirement would be to have at least one transition every eight bits. This would mean that an eight bit pattern would have to be inserted in place of any eight bit sequence of the same value. The selection of the particular pattern to be used for insertion must be chosen in a manner similar to that of a synchronization pattern for frame sync. The insertion sequence must meet the following:

1. The probability of it occurring naturally in the data stream must be extremely small. Since it is very desirable to avoid false recognition at the receiver, which would result in a misinterpretation of valid data for inserted data.
2. The inverse of the sequence must also be available. Since the bit sequence of the same value may be either ones or zeros, two separate sequences are needed.

3. The effects of channel errors must be considered. Should the ground receiver accept no errors, one error, two errors, etc? If a short sequence such as the eight bit one used in the example is selected, then the no error case would probably be best. The probability of no errors in any 8 bits is .99992 thus the probability of not recognizing the inserted pattern due to channel error would be about 8×10^{-5} , slightly greater than the channel error rate of 1×10^{-5} . In the one error case the probability of false synchronization detection would be much higher than the gain in recognizing a valid insertion with a single error.

By employing this simple insertion technique, one does not take advantage of the natural transition that might occur prior to and following the sequence of the same value. As stated above the timing and clocking circuitry would be complex even for this simple case. For this reason bit insertion is not the most favorable method although it can be designed to meet the system constraints given in Chapter 2. But it should be noted that only the replacement type can be employed. The inserting of additional bits would decrease the information rate and therefore this type is not acceptable. Since the amount of additional bits cannot be predetermined.

3.B.4 Error Encoding

Telecommunication systems often employ different types of error correcting codes to improve their bit transition density. In this manner, the error correcting codes provided two services. First, they improve the channel bit error rate and second, they provide an increase transition density. This type of method is employed by the 50 Mbit channel of the shuttle as well as by many other systems. All these systems add additional bits to the data and thus decrease the information rate. Therefore they are not viable candidates for the HBTB modification, but are included in the following for completeness.

The following discussion concerning the output symbol transition density of the 1/3 convolutional encoder with alternate symbol inversion is based on the material by M.K. Simon and J.F. Smith in Reference 6 and

illustrated in Figure 3.2. Simon and Smith have determined, for a particular class of convolutional codes, that alternate symbol inversion assures a maximum transition-free run of output symbols, and hence its minimum transition density. This maximum length is independent of the data source model, independent of the code connections, and dependent only on the code constraint length and rate. Simon and Smith separate all $1/v$ convolutional codes into three classes of codes: V_{even} , V_{odd} for transparent codes, and v odd for nontransparent codes. A transparent code is one which provides the complement of the output sequence for the complement of the input sequence. A simple test to determine if a code is transparent is each row of the generator matrix \underline{C} has an odd number of ones then the code is transparent.

The generator matrix \underline{C} for the $1/3$ convolutional code employed by the Space Telescope (ST)

$$\underline{C} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (3.3)$$

where the right hand column represents the present input and the left hand column represents the oldest (content of the last shift register K , the code constraint length) input.

Since $v = 3$, odd and each row of \underline{C} contains an odd number of ones, the convolutional code is a member of case 2. Simons and Smith state for v odd and transparent codes, the only input bit sequence that will produce an output alternating sequence longer than N_{max} symbols, where N_{max} is defined as

$$N_{\text{max}} = K + \left\lceil \frac{K-1}{v-1} \right\rceil - 1 + v \quad (3.4)$$

K = the code constraint length; $\lceil X \rceil$ denotes the smallest integer greater than or equal to X .

is the alternating sequence. Furthermore, if the encoder is such that the alternating input sequence produces the alternating output sequence, then this output sequence can continue indefinitely, i.e., alternate symbol inversion will not produce a finite transition-free symbol sequence.

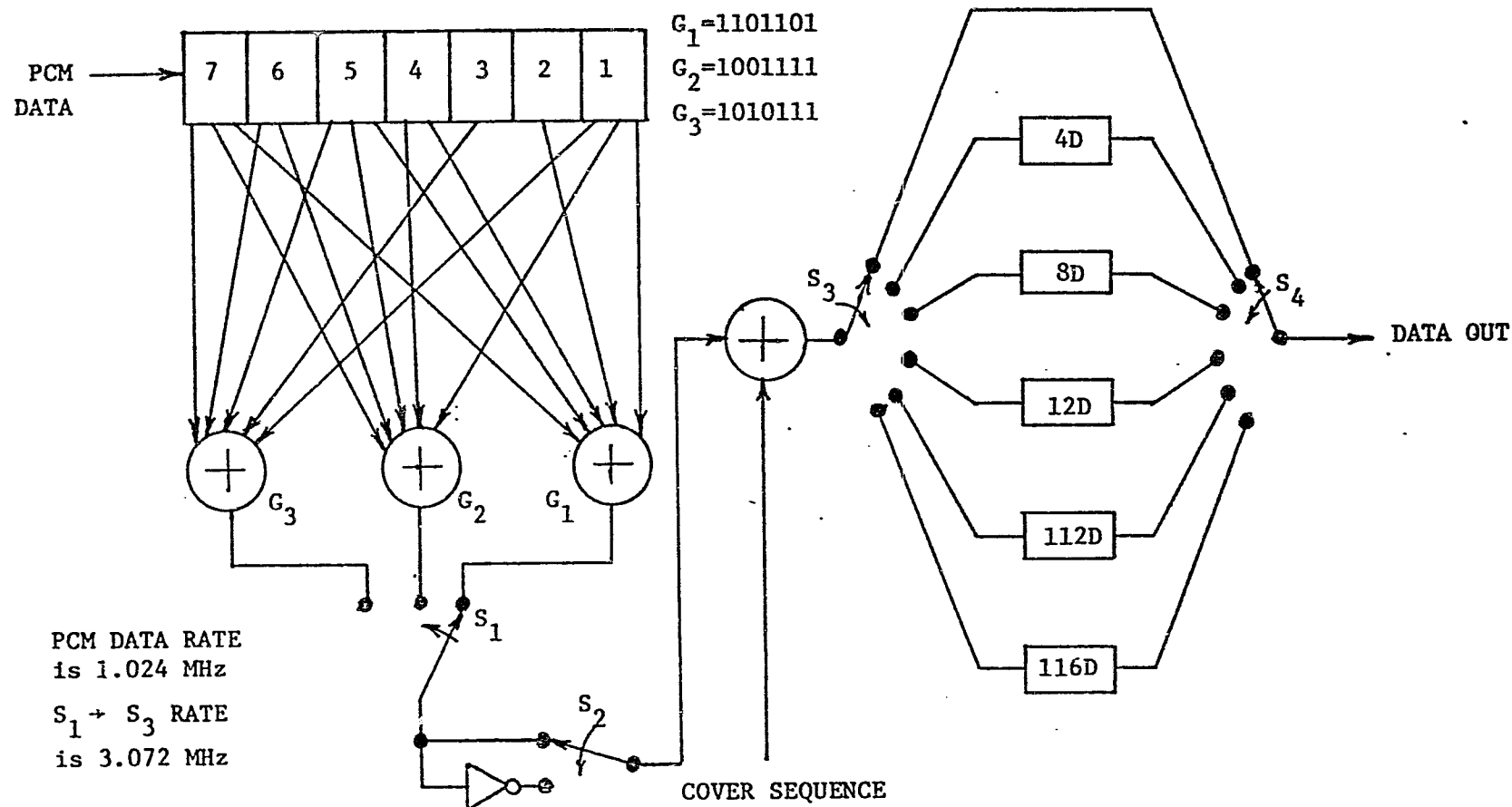


Figure 3.2 Convolutional Encoder with PCI and Cover Sequence

ORIGINAL PAGE IS
 OF POOR QUALITY

Reference 6 provides a test to determine if a case 2 code will produce an alternating output for an alternating input. Split the generator matrix \underline{C} into two matrices $\underline{C}_{\text{odd}}$ and $\underline{C}_{\text{even}}$ where $\underline{C}_{\text{odd}}$ is composed of all the odd columns of \underline{C} and $\underline{C}_{\text{even}}$ all the even columns. If the number of ones in each row of the matrix formed by stacking $\underline{C}_{\text{odd}}$ on top of $\underline{C}_{\text{even}}$ alternates even, odd, even, ... or vice versa, then an alternating input sequence will produce an alternating output sequence. Testing the generator matrix, it is found the number of ones in each row of the test matrix does not alternate even, odd or vice versa. Therefore the maximum number of transition-free output symbols from the 1/3 convolution encoder with alternate symbol inversion is

$$\begin{aligned}
 N_{\text{max}} &= K + \left\lceil \frac{K-1}{v-1} \right\rceil - 1 + v \\
 &= 7 + \left\lceil \frac{7-1}{3-1} \right\rceil - 1 + 3 \\
 &= 12.
 \end{aligned} \tag{3.5}$$

The maximum number of transition-free output symbols was also determined to be 12 in References 7 and 8. Magnavox in Reference 7 utilized an extensive computer analysis to arrive at a maximum of 12. Baument, et al., Reference 8, used a slightly different mathematical approach to obtain 12 as the maximum bits between transitions and therefore the system is guaranteed to meet the 1 in 64 requirement.

Simon and Smith also prove in Reference 6 the 11-bit input sequence 01110100100 yields the output 010000000000001. Neither this output sequence nor its compliment can be repeated within the next 33 output symbols. The next input will produce at least one additional bit transition therefore the average bit transition for this worse case plus one additional input is 2 transitions per 16 output symbols which yields an average of 1 transition every 8 output symbols. Therefore the output of the 1/3 convolutional encoder with alternate bit inversion and generator matrix given in Equation 3.3 will meet both the 1 transition per 64 bits and 64 transitions in 512 or an average of 1 transition every 8 bits.

If the 12-bit input sequence is 01110100100 the output will be 010000000000001100.

If the 12-bit input sequence is 01110100101 the output will be 010000000000001011.

Since the output of the 1/3 rate convolutional encoder will have a transition at least every 13 bits independent of the data input, it is not necessary to examine the equipment preceeding the encoder. However if the channel interleaver is utilized it is necessary to determine if it is possible to obtain 64 or more symbols out of the interleaver without a transition. The channel interleaver is shown in Figure 3.3. This interleaver will take any two symbols within 30 of each other and separate them by at least 119 bits. Equation 3.6 may be used to express a typical output symbol b_i in terms of the input symbols a_i .

$$b_{j+i} = a_{j+i-4 \times 30} = a_{j+i-120_i} = a_{j-119_i} \quad j \geq 119_i \quad (3.6a)$$

$$b_{j+i} = 0 \quad j \leq 119_i \quad (3.6b)$$

where

$$j = 0, 30, 60, 90, 120, \dots$$

$$i = 0, 1, 2, 3, \dots, 29$$

Therefore, a typical output sequence of the interleaver would resemble a sampling of the input sequence with the samples being taken every 119th bit for sequences up to 30 bits in length. In order for the interleaver to have an output of 64 consecutive symbols of the same value, the input data must be such that samples of the input sequence separated by 119 symbols be of the same value. The length of input symbols corresponding to 64 output symbols is approximately 3511. Also noting that the output of the interleaver is combined with a PN cover sequence of length 30, it would appear highly unlikely that a string of 64 ones or zeros will occur, however due to the systematic construction of the components of the system, it is possible that a

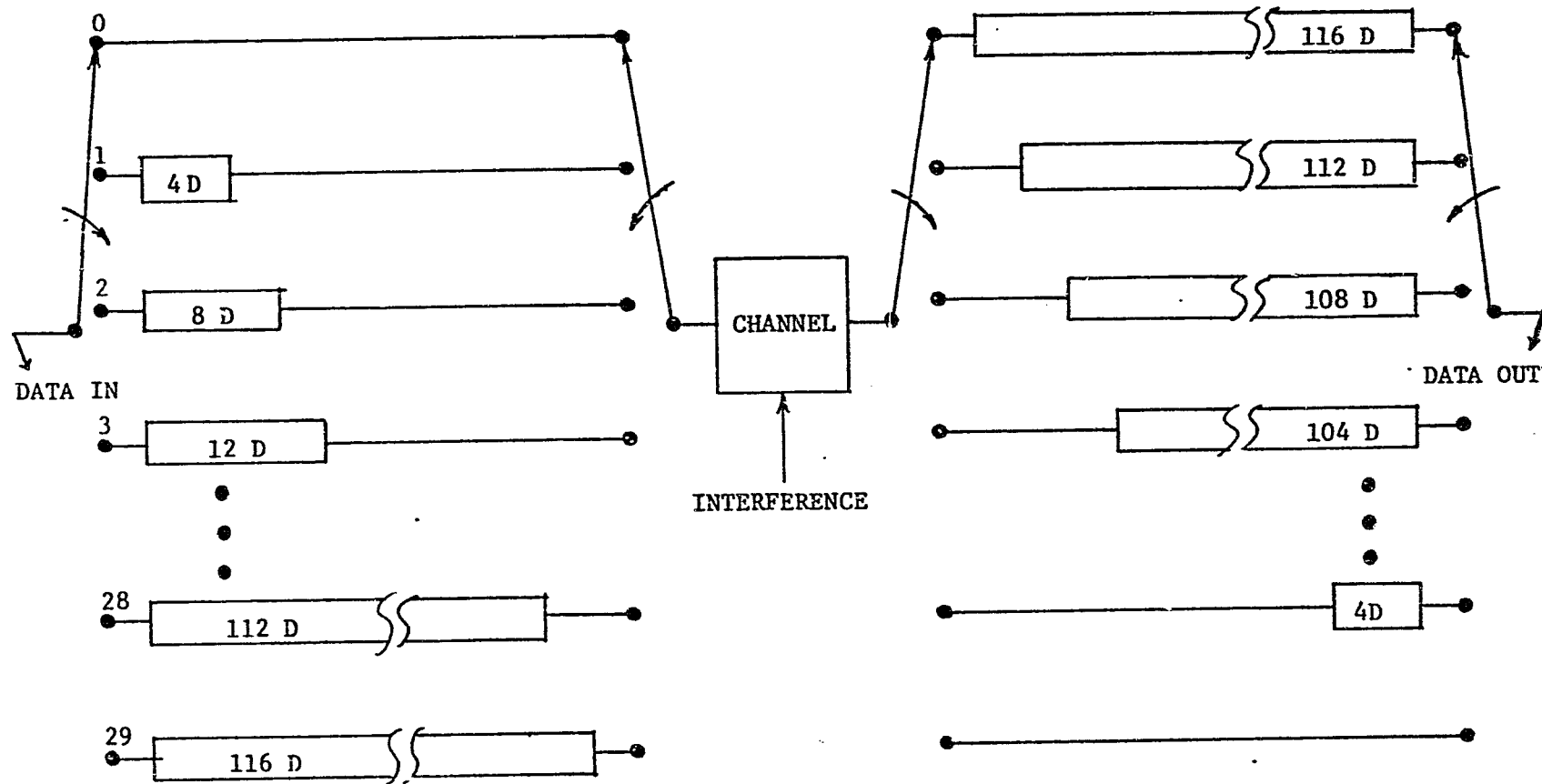


Figure 3.3 Periodic Convolutional Interleaver and Deinterleaver

ORIGINAL PAGE IS
OF POOR QUALITY

sequence of data does exist that will yield a string of 64 output symbol without a transition. Since the actual structure of the data is presently unavailable it is not possible to examine this problem more closely. It would be necessary to examine very closely the structure of the data and how that structure is effected by the various components of the system.

3.B.5 Bit Scramblers

A bit scrambler is a digital machine which maps a data sequence into a channel sequence and with the special mapping of a periodic data sequence into a periodic channel sequence with period much greater than the data period. For periodic source, the channel sequence produced by the scrambler, also, has many transitions.

The basic element of all scramblers is a feedback shift register generator (FBSRG) with tap polynomial $h(x)$; where $h(x)$ is a primitive polynomial over the field $GF(p)$, p is prime. The manner in which this element is connect determines whether the scramblers is self-synchronizing or not. The self-synchronizing group utilizes the data sequence to drive the FBSRG. The non-self synchronizing group, often call reset, utilizes the FBSRG as a maximal length (ML) generator and modulo adds the ML sequence to the data. Each group is discussed in the following paragraphs.

3.B.5.a Self-Synchronizing Scramblers

The self synchronizing group maybe subdivided into two types called multi-counter scramblers (MCS) and single-counter scramblers (SCS). Both types consists of a "basic self-sync scrambler" and a "monitoring logic". Figure 3.4 illustrates the "basic self-synchronizing scrambler" (BSS). The logic circuit determines the scrambler type Figures 3.5 and 3.6 show the MCS and SCS respectively.

The BSS when excited by a periodic sequence of period* 's' will respond with a periodic line sequence which has either period 's' or a period which is the least common multiple (LCM) of 's'

*A sequence has period 's' if it is the smallest period in the sequence.

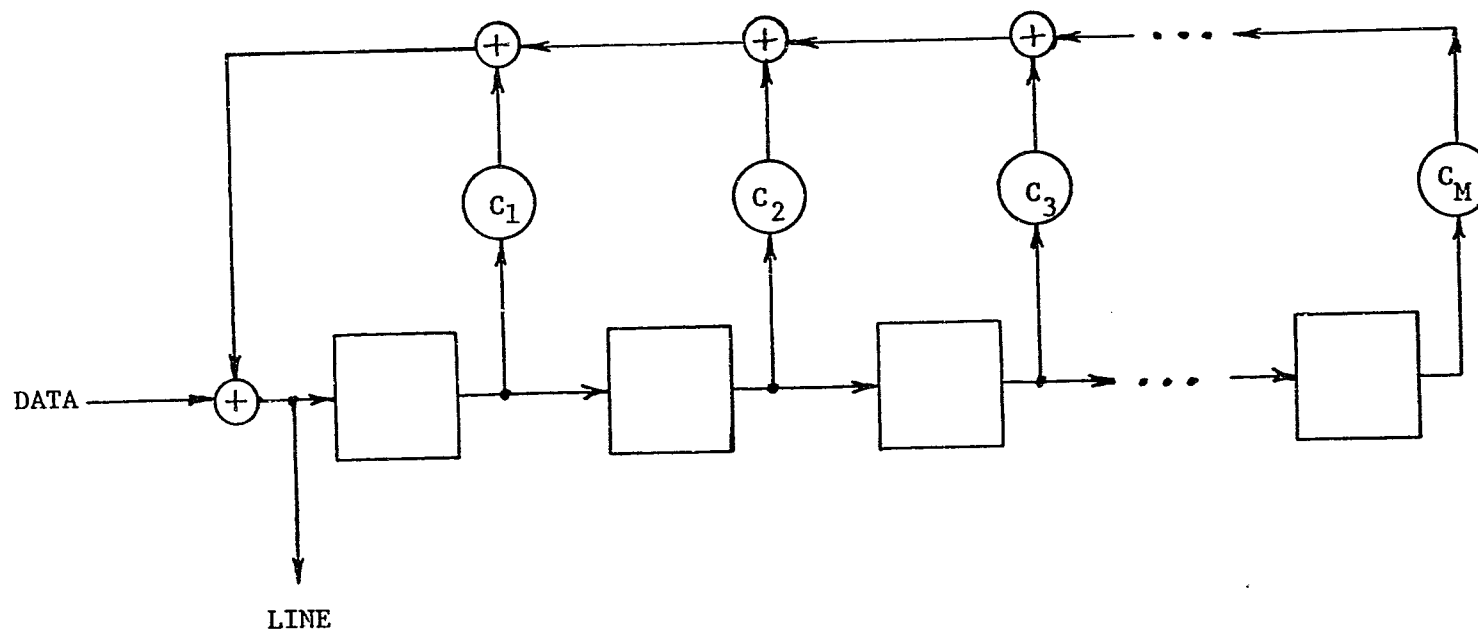


Figure 3.4 General Basic Self-Sync Scrambler

$$h(x) = x^m + C_1 x^{m-1} + C_2 x^{m-2} + \dots + C_m$$

where $C_i = 1$ or 0 and $h(x)$ is a primitive polynomial over $GF(2)$ of degree M . C_m must equal one.

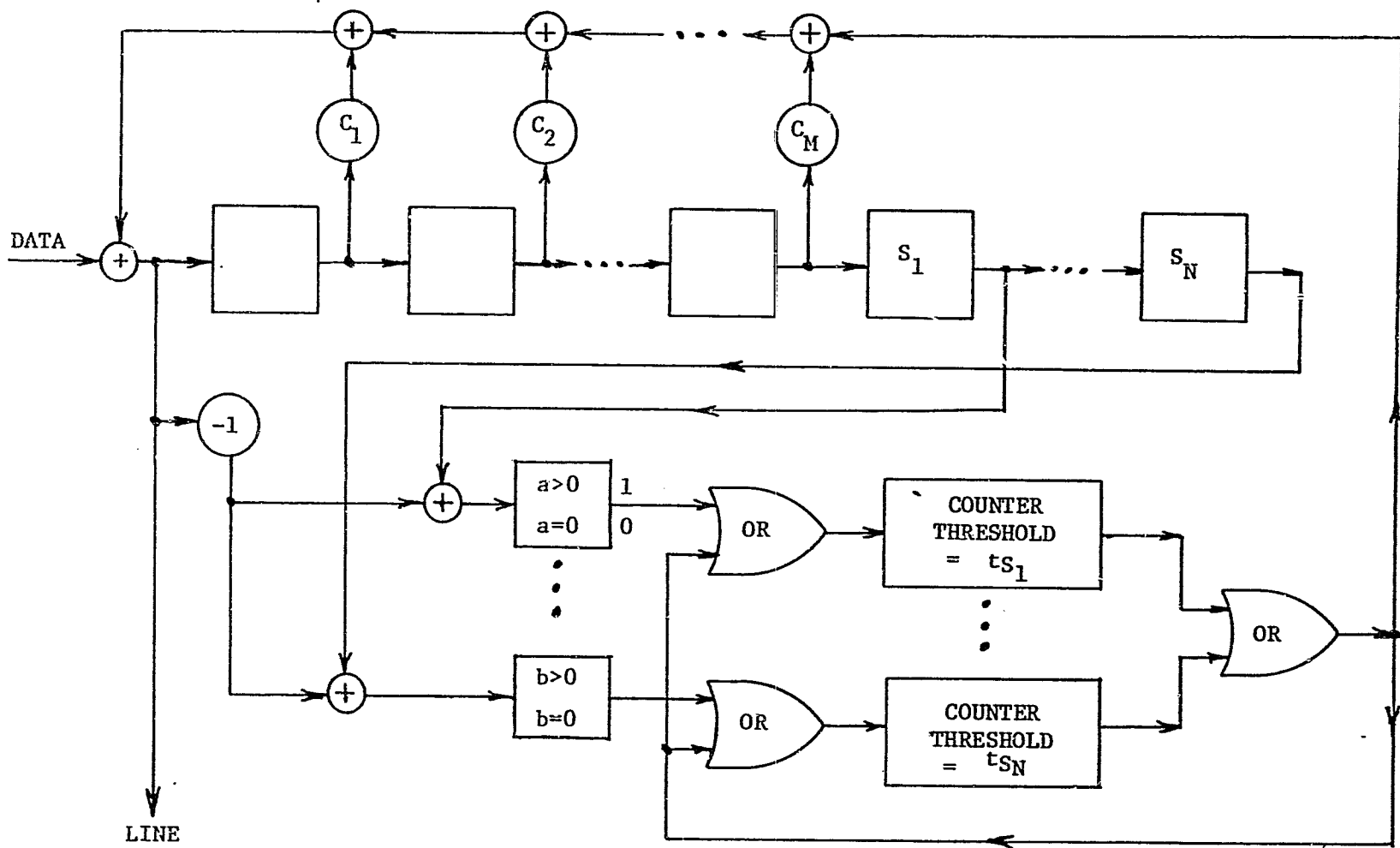


Figure 3.5 Multi-Counter Scrambler

ORIGINAL PAGE IS
OF POOR QUALITY

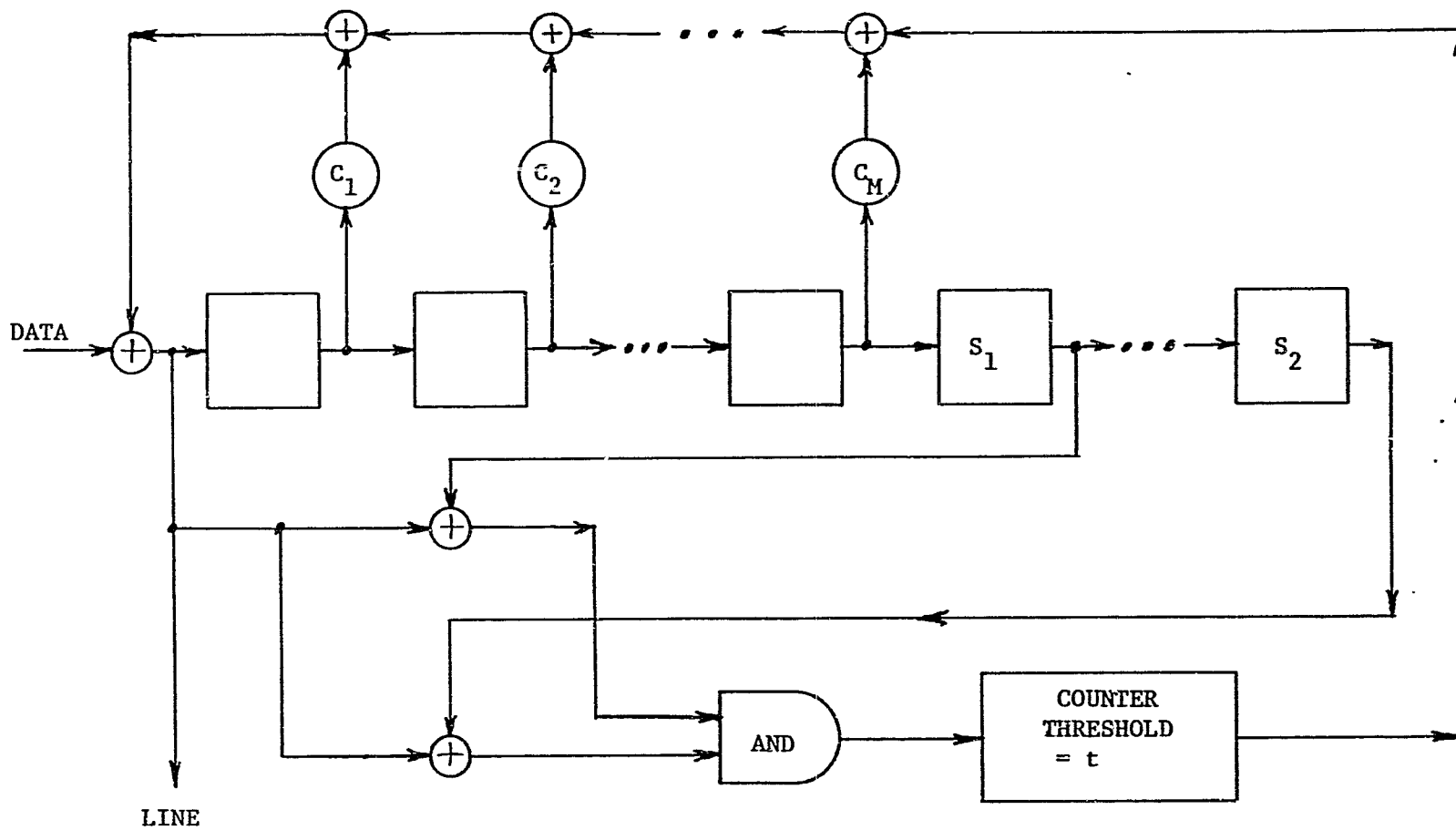


Figure 3.6 Single Counter Scrambler

ORIGINAL PAGE IS
OF POOR QUALITY

and p^{m-1} * (denoted by $\text{LCM}(s, p^m-1)$). The period with which the scrambler responds is a function of the initial values stored in the scrambler storage elements, (its initial state) and there is only one such state (for each phase of input sequence) for which the line sequence has period 's'. For all other such initial states the line sequence has the larger period. The preceding statements are Savage's Theorem 1 for BSS (See Reference 9 for proof).

The logic circuit employed by the MCS and SCS are used to detect the presence of a periodic sequence of low period on line and alter the starting state of the BSS to insure the line sequence has period of $\text{LCM}(s, p^m-1)$.

3.B.5.a.1 Multi-Count Scrambler

The logic used by the MCS is more general than the SCS and allows for the simultaneous detect of sequences of several periods. The MCS employs N counters, one for each period ' s_i ', $1 \leq i \leq N$, and the i^{th} counter will generate +1 if it reaches its threshold t_{s_i} . The counter is reset whenever the reset lead is nonzero so that t_{s_i} consecutive zeros on the reset lead of the i^{th} counter will cause it to reach its threshold. Whenever a counter reaches its threshold a '1' is added to the feedback line of the BSS, thereby change the state of the BSS. Thus, the line sequence will then be changed from period ' s_i ' to period $\text{LCM}(s_i, p^m-1)$ where the i^{th} counter was the one reaching threshold. At the same time, all counters are reset.

Thus, the MCS shown in Figure 3.5 will scramble a periodic sequence of period 's' if 's' divides ' s_i ' (denoted by ' s_i/s ') for some i, $1 \leq i \leq N$, and will produce a periodic line sequence of period $\text{LCM}(s_i, p^m-1)$ if the following two conditions are met:

1) The tap polynomial $h(x)$ of degree m is primitive over $\text{GF}(p)$ where data sequences have components from $\text{GF}(p)$.

2) The thresholds t_{s_i} , $1 \leq i \leq N$ are chosen as

* p^m-1 is the period of the maximal length sequence generated by BSS in the absence of an input.

$$t_{s_1} \geq (m-1) + \max_{\substack{1 \leq j \leq N \\ j \neq 1}} 's_j' . \quad (3.7)$$

If all input periods divide ' s_0 ', then the statement holds when condition (i) is met and a threshold of $t_{s_0} \geq m$ is used. The above is Savage's MCS theorem. (See Reference 9 for proof.)

3.B.5.a.2 Single-Counter Scramblers

The SCS is designed to scramble periodic binary sequences whose periods divide either ' s_1 ' or ' s_2 ' or both. Since the SCS utilizes only one counter, it may be less costly to build than the MCS in some case. The SCS operates in the same manner as the MCS.

Savage's SCS theorem states that a SCS exists which will scramble all periodic binary sequences with periods which divide ' s_1 ' or ' s_2 ' where $s_1 < s_2$ and s_1 does not divide s_2 (denoted by $s_1 \times s_2$) if

- 1) the tap polynomial $h(x)$ of degree ' m ' is primitive over $GF(2)$,
- 2) s_1 and s_2 are relatively prime to $2^m - 1$, and
- 3) a counter threshold, t , $t \leq s_2(2^m - 1) - 2^{m-1} + 2$ is chosen.

See Reference 9 for proof.

3.B.5.a.3 Transition Density for Self-Synchronizing Scramblers

Transitions occur frequently in a scrambled periodic sequence and in one period of a scrambled sequence there are approximately half as many transitions as there are digits. These have been illustrated in Reference 9 when the source is binary and the scrambler input periods are relatively prime to $2^m - 1$, where m is the size of the BSS.

Assuming the BSS generates a line sequence with period ' ℓ ' when the input has period ' s ', the source is binary, the BSS has ' m ' stages, and ' s ' is relatively prime to $2^m - 1$; then ' ℓ ' is an ' $s(2^m - 1)$ ' component vector. If the binary line sequence is converted into a line signal by the mapping $1 \rightarrow +1$, $0 \rightarrow -1$ and if it is linear modulated,

then Savage's transition theorem states "The binary vector ' ℓ ' of length $s(2^m-1)$ representing the response of a binary scrambler to an input of period ' s ', when ' s ' and 2^m-1 are relatively prime, has at least one transition every ' s ' + ' m ' digits and has a total of $\text{Tr}(\ell)$ transitions where

$$\frac{1}{2} \left[\frac{2^m-2}{2^m-1} \right] \leq \frac{\text{Tr}(\ell)}{s(2^m-1)} \leq \frac{1}{2} \left[\frac{2^m}{2^m-1} \right] \quad (3.8)$$

This theorem may hold for reset scrambler also, but Savage's proof does not take into count the reset scrambler. Therefore prior to applying these bounds to the reset scrambler further evaluation is needed.

3.B.5.a.4 Self-Synchronizing Descramblers

The descramblers for the MCS and SCS are shown in Figures 3.7 and 3.8, respectively. The descrambler is said to be out of synchronism with the scrambler if either (1) the values in the BSS and the delay elements differ from those stored in the corresponding sections of the scrambler or (2) if the counters in the monitoring logic are not at the same levels as those at the scrambler or both. Examining Figure 3.7 or 3.8, it can be seen that the delay elements of the descrambler will be purged after ' s_N ' clock intervals, provided ' s_N ' is the largest expected period (number of delay elements). The monitoring logic at the scrambler and descrambler will be at the same level after an additional ' s_N ' clock intervals provided no line errors have occurred. Therefore the descrambler will require at most $2 \times 's_N'$ clock intervals free of error (channel errors or bit slip) to recover sync.

The primary effect of a channel error on the descrambler is to introduce additional errors. If the effect on the monitoring logic is neglected, the descrambler will produce approximately $w(h)$ as many output errors as channel errors, where $w(h)$ is the number of non-zero terms in the tap polynomial $h(x)$.

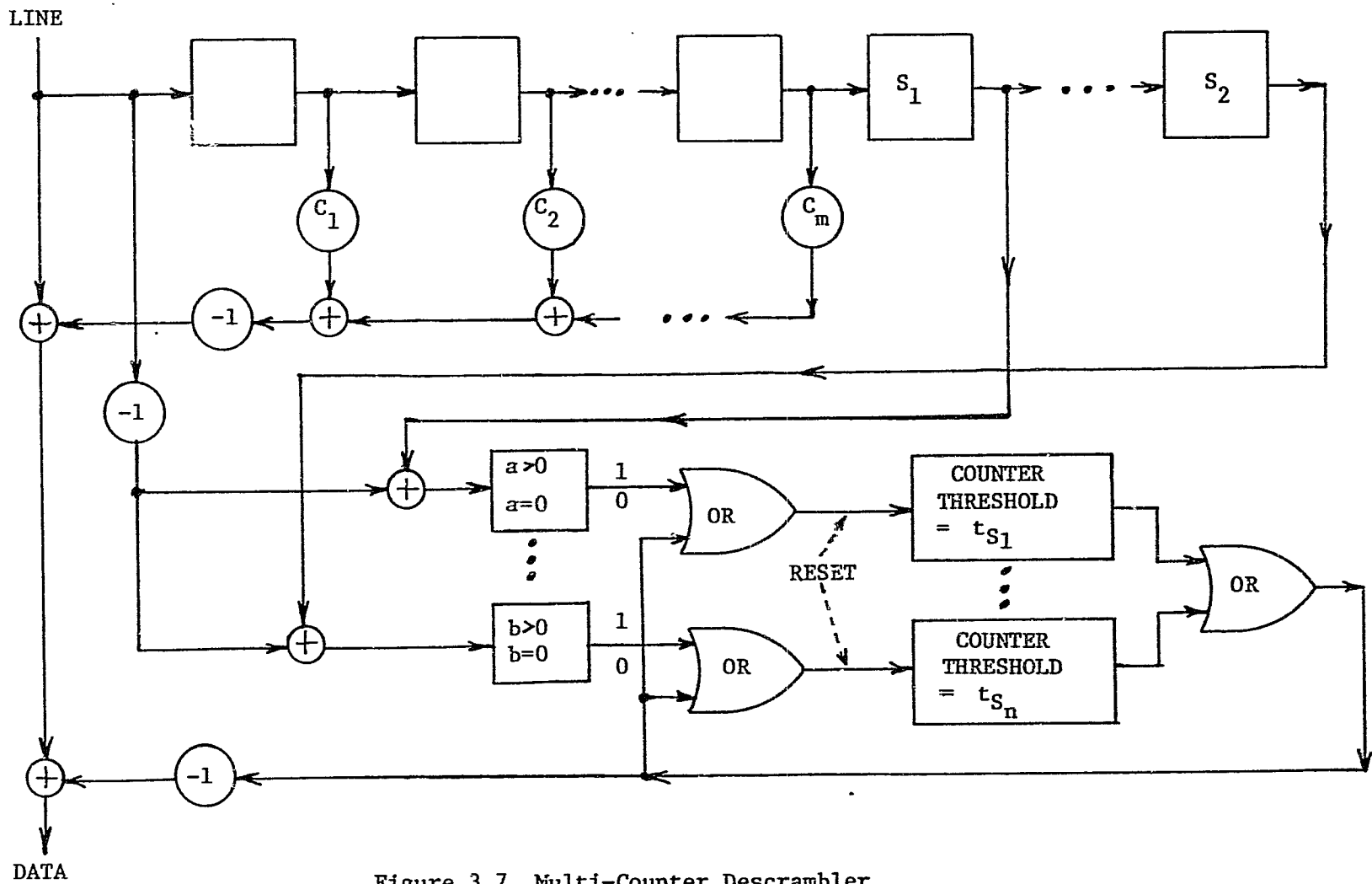


Figure 3.7 Multi-Counter Descrambler

ORIGINAL PAGE IS
OF POOR QUALITY

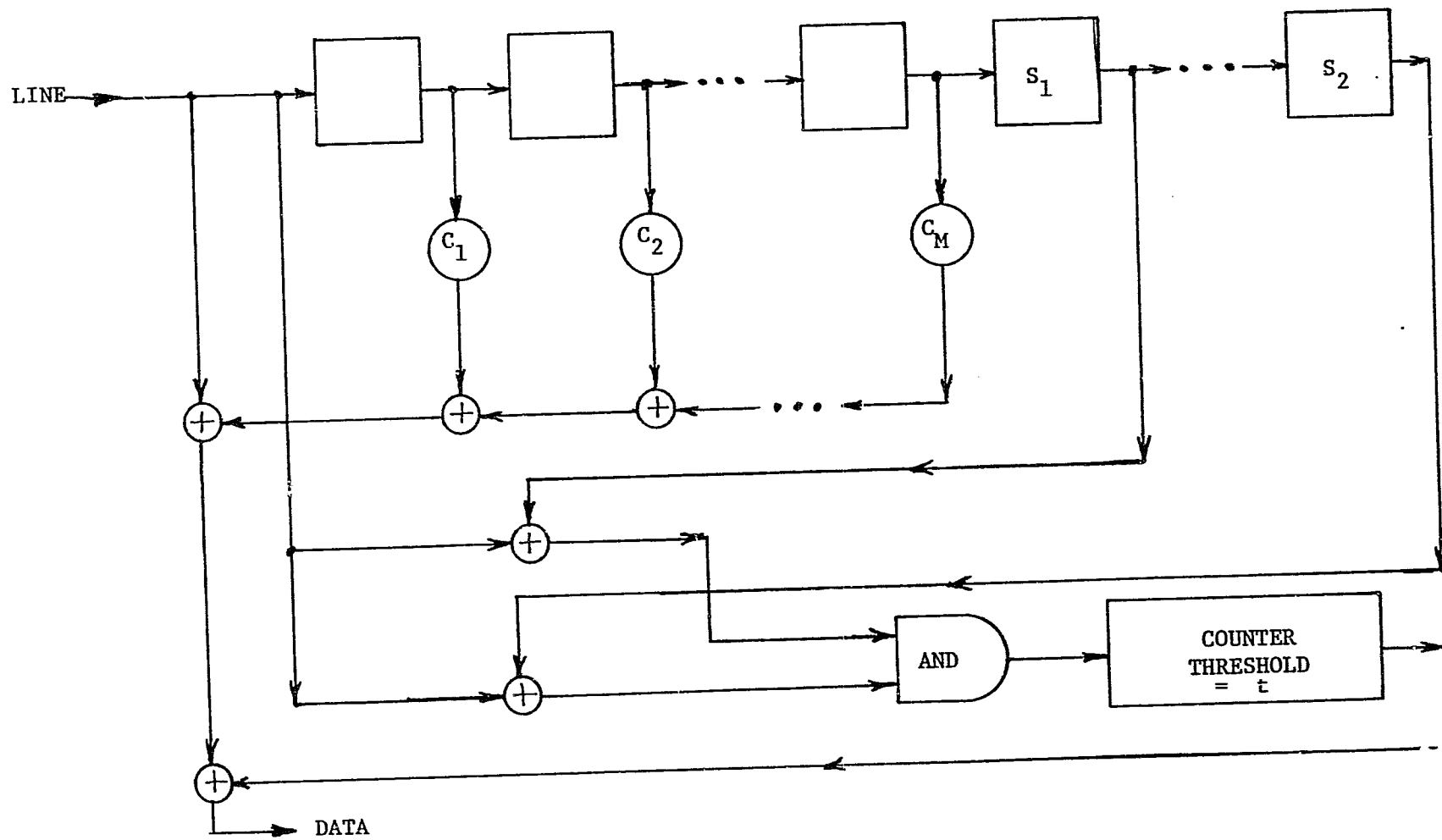


Figure 3.8 Single-Counter Descrambler

ORIGINAL PAGE IS
OF POOR QUALITY.

3.B.5.a.5 The Spectrum of the Scrambler Output

Assume a linearly modulated carrier, a binary source, and the source is converted into a waveform such that $0 = -1$, $1 = +1$. Let T_0 be the time interval allotted to each binary digit and let $\hat{\ell}(t)$ be the waveform generated by the binary sequence ℓ .

If ℓ is the output of the scrambler for an equiprobable, independent source input, then ℓ is a sequence of independent, equiprobable, binary digits. Thus the autocorrelation function of $\ell(t)$ is

$$R_{\ell}(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_0} & , \quad |\tau| \leq T_0 \\ 0 & , \quad |\tau| > T_0 \end{cases} \quad (3.9)$$

and the power density spectrum for $\hat{\ell}(t)$ is

$$S(f) = T_0 \left(\frac{\sin \pi f T_0}{\pi f T_0} \right)^2 \quad (3.10)$$

Now let the source be periodic, with period 's' such that the line sequence has period $T_0(\text{LCM}(s, 2^m-1))$; then the power density spectrum for $\hat{\ell}(t)$ is

$$S(f) = \frac{1}{P} \delta(f) + T_0 \left(\frac{\sin \pi f T_0}{\pi f T_0} \right)^2 \left\{ \frac{u}{sPT_1} \sum_{j=-\infty}^{\infty} \delta\left(f - \frac{j}{PT_1}\right) + \left(1 - \frac{u}{s} - \frac{1}{P}\right) \frac{1}{PT_0} \sum_{j=-\infty}^{\infty} \delta\left(f - \frac{j}{PT_0}\right) \right\} \quad (3.11)$$

where $P = 2^m - 1$

$$T_{1..} = sT_0$$

u is a function of the scrambler input (the number of 1's in $\ell + \ell_k$, k a multiple of P , depends on the input; ℓ_k represents k cyclic shifts of ℓ).

In other words the principle effect of scrambling is to decrease the number of tones in a given bandwidth by a factor which is approximately P and to decrease the level of each tone by approximately the same factor. The bandwidth is unchanged.

3.B.5.a.6 Serial, Cascaded, and Parallel Scramblers

Self-synchronizing scramblers may be classed in subgroups depending on the inter connects between the scramblers. A single scrambler with m delays is called a serial scrambler. The scramblers described in the above were serial.

Scramblers may also be connected in cascaded, meaning the output of one is fed to the input of the next scrambler. Cascade scramblers are inferior to serial scramblers with the same number of delay elements. For example 4 serial scramblers connected in cascade (see Figure 3.9) has a longest output period (output of the 4th scrambler) of only the $\text{LCM}(s, 4(2^m - 1))$ and a probability of occurrence of $(1 - 2^{-2m})$ where as a single scrambler with $4m$ delays has a maximum output period of $\text{LCM}(s, 2^{4m} - 1)$ with a probability of occurrence of $(1 - 2^{-4m})$. Therefore a serial scrambler is preferred to a cascaded scrambler.

Scramblers may also be connected in parallel. The main advantage of parallel scramblers is the reduction of the number of output errors to input errors. Parallel scramblers use two inputs thereby requiring additional components for series input sequences. Figure 3.10 shows the configuration for parallel scramblers. Examining Figure 3.10, if an error occurs on input 'a', $w(h) + 1$ errors will be produced by the output, but if an error occurs on input b only one will be produced in the output. Therefore, in the case of parallel inputs, the output errors will be reduced for line b. This technique provides little improvement for random errors occurring in serial data; however.

References 9 and 10 provide additional information on self-synchronizing scramblers. The main point concerning self-synchronizing scramblers is that the principal effect of infrequent channel errors on the descrambler is to multiply the number of channel errors by $w(h)$, where $w(h)$ is the number of non-zero terms in the tap polynomial $h(x)$.

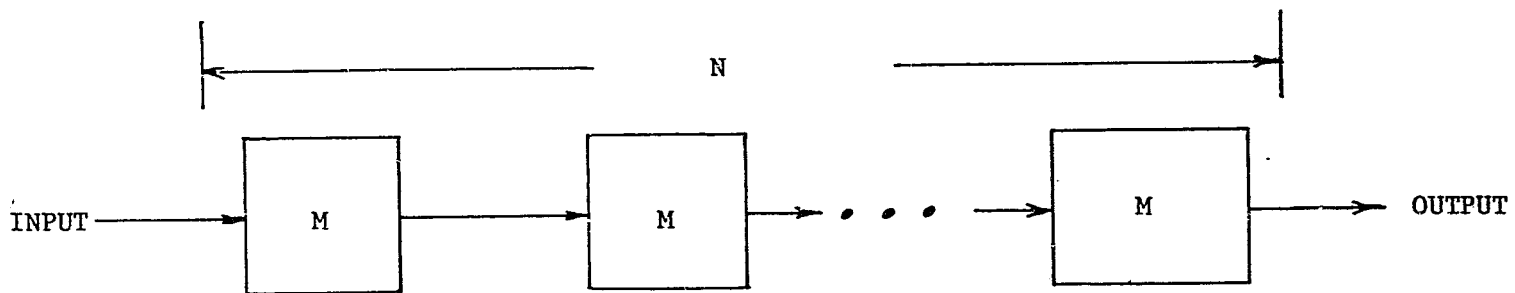


Figure 3.9 Cascading of N M-bit Scramblers

ORIGINAL PAGE IS
OF POOR QUALITY

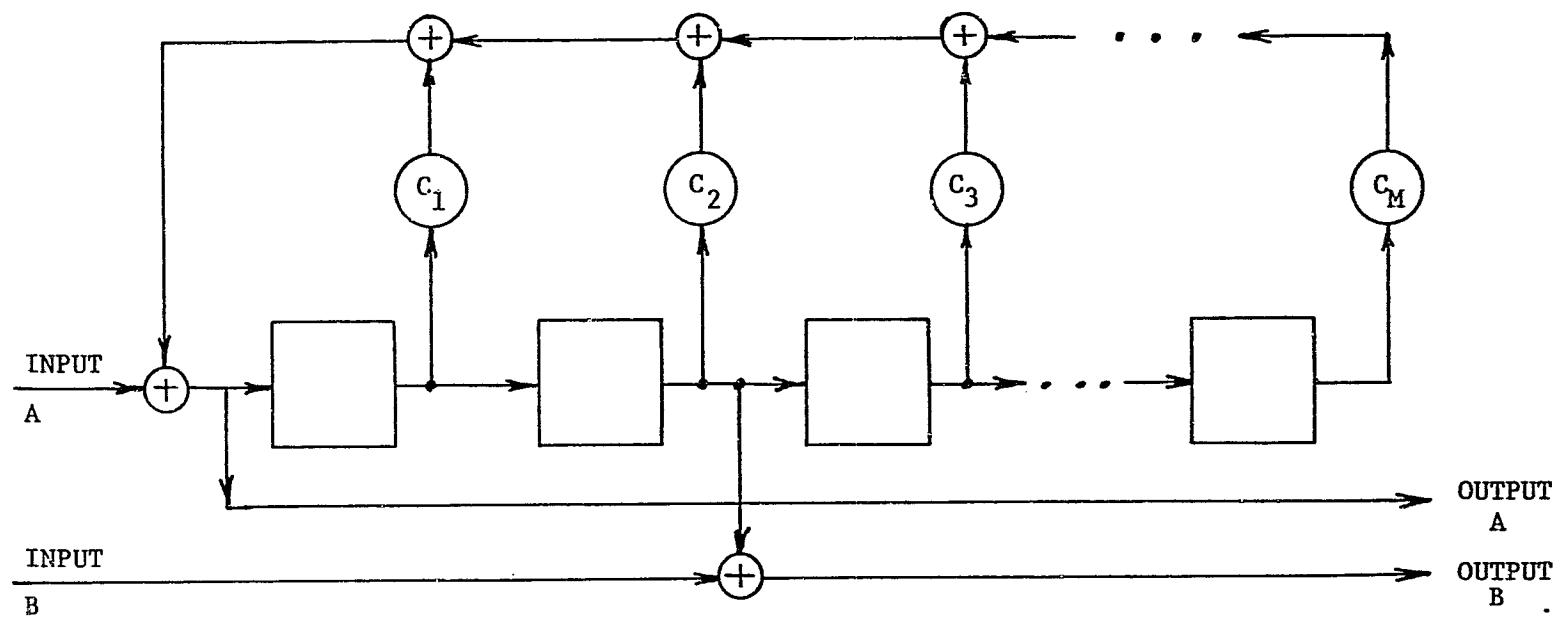


Figure 3.10 Parallel Scrambler Configuration
(Monitoring Logic Omitted)

3.B.5.b Non-Self Synchronizing (Reset) Scramblers.

The reset scrambler is simple, consisting of a maximal-length sequence generator modulo 2 added to the data sequence prior to modulation. The reset scrambler must be provided with synchronizing pulses in order to relock on sync. Since the ML sequence is independent of the data stream there is no multiplication of output errors relative to input errors.

At present the number of transition for a given sequence length is unknown. It is possible that the reset scrambler may provide the same $Tr(\ell)$ as the self-synchronizing scrambler, but the proof utilized by Savage⁹ cannot be directly applied to the reset scrambler.

The scrambler and descrambler for the reset group are identical maximal-length sequence generator for binary data. Figure 3.11 shows the reset scrambler. The synchronizing pulse may be obtained from the frame sync or another sync pattern found in the data stream. Since the sync pattern would be utilized to reset and start the ML-sequence, it would not be scrambled.

The self-synchronizing scramblers are poor candidates for the SL due to the multiplication of channel errors. The reset group is a better candidate provided a lower bound can be determined for the number of transitions for a given sequence length, and is discussed in the next Chapter. (See Reference 12 for a brief summary on scramblers.)

Table 3.1 summarizes the possible methods discussed in this section for improving bit transition density of the 2 Mbit SL Link. Only the reset scrambler remains as a viable option. The reset scrambler is actually a PN Cover Sequence which is modulo-2 added to the data stream. This technique is examined in further detail in Chapter IV.

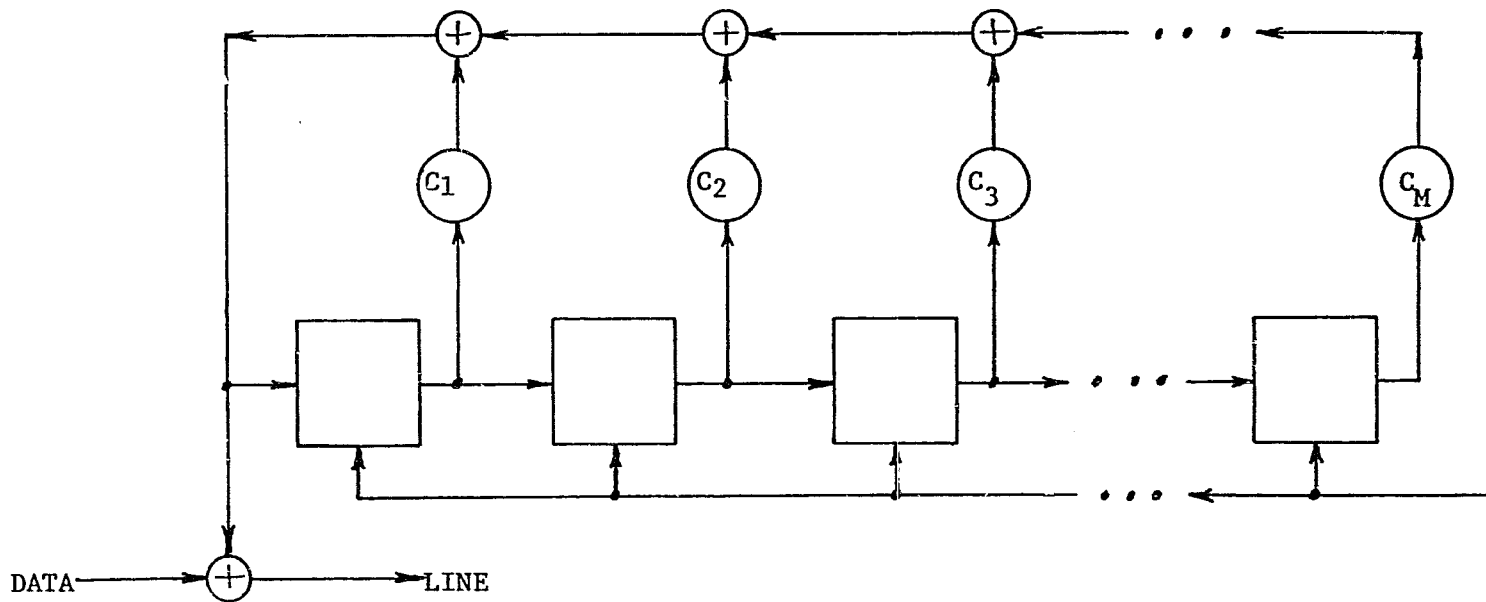


Figure 3.11 Reset Scrambler and Descrambler

$$h(X) = X^m + C_1 X^{m-1} + \dots + C_m$$

where $C_i = 1$ or 0 and $h(X)$ is a primitive polynomial over $GF(2)$ of degree M

TABLE 3.1 BIT TRANSITION DENSITY CODING CHOICES

TECHNIQUE	SPECTRAL BW	ERROR CHARACTERISTICS	SYNCHRONIZATION	HARDWARE COMPLEXITY	TRANSITION PROPERTIES	CONSTRAINTS VIOLATED
1. Bit Scrambler Self Synchronizing						
a. Serial	Same as Data Stream	Error propagation and multiple output errors for single error input	Self Synchronizing	Encoder-Moderate	A Guaranteed Performance can be Provided	#3
b. Parallel	Same as Data Stream	Same as 1a.	Self Synchronizing	Encoder-Moderate	A Guaranteed Performance can be provided	#3
2. Delay Modulation						
a. Miller	Same as Data Stream	Double errors out for single error in	Must Provide Synchronization	Encoder-Moderate Decoder-More Complex	2 Transitions in every 3 bit cells	#3
b. Miller ²	Same as Data Stream	Double errors out for single error in	Must Provide Synchronization	Encoder-Moderate Decoder-More Complex	1 Transition in every 3 bit cells	#3
3. Bi-Phase Modulation (Manchester, Manchester II, Bi-Phase Mark, Bi-phase Space)	Double Data Bandwidth	1 error out for 1 error in	Self Synchronizing	Encoder-Moderate Decoder-Moderate to Complex	1 transition for every bit cell	#2
4. Alternate Bit Version	Same as Data	1 bit slip yields inverted bit stream. 1 error out for 1 error in.	Must Provide Synchronization	Encoder-Moderate Decoder-Moderate	Data Pattern Sensitive. No Guarantees. Statistically Good	#3
5. Sequence Insertion (Includes Error Codes)	Same as Data	Possible error multiplication due to channel errors and data convolutions.	Must Provide Synchronization Sensitive to Inverted Data Stream	Timing and Counting Complex	Can be Designed to Meet Specifications.	#3
6. PN Cover Sequence (Reset Scrambler)	Same as Data	1 error out for 1 error in.	Must Provide Synchronization	Encoder-Simple Decoder-Moderate	Can be Designed to Meet Specifications.	NONE

ORIGINAL PAGE IS
OF POOR QUALITY

CHAPTER 4

PSEUDO-NOISE COVER SEQUENCE (Reset Scrambler)

In the preceding chapter, different techniques were examined to determine their capability of resolving the bit transition density problem of the SL 2 Mbit Return Link and these techniques are summarized in Table 3.1. Only the PN Cover Sequence (or Reset Bit Scrambler) was capable of meeting all the system constraints listed in Tables 2.1 and 2.2. This chapter deals primarily with the particular PN sequence chosen for the SL, however a brief general discussion of PN sequences is also given.

4.A PSEUDO-NOISE (PN) SEQUENCES

Pseudo-Noise Sequences are binary-valued, noise-like sequences in that they are purely random. That is any bit in the sequence may be a one or a zero with equally likely probability. However their primary advantages are that they are deterministic, easily generated by feedback shift registers, and they have a correlation function which is highly peaked for zero delay and approximately zero for other delays. By proper selection of the tap polynomial, which indicates the feedback connections, a maximal linear (ML) sequence or m-sequence is generated. Figure 4.1 illustrates a general ML sequence generator. A ML sequence has a length, $L = 2^n - 1$, where 'n' is the number of stages in the shift register (SR). The number of ones in the sequence equals the number of zeros plus one. There are

$$\left[\frac{2^n - 1}{2} \right] \text{ zeros and } \left[\frac{2^n - 1}{2} \right] + 1 \text{ ones.}$$

The number of transitions within the sequence is approximately half the number of bits in the sequence. The number of transition equals

$$\left[\frac{2^n - 1}{2} \right]. \text{ The maximum number of bits without a transition is}$$

equal to the number of stages in the SR, 'n'. The statistical

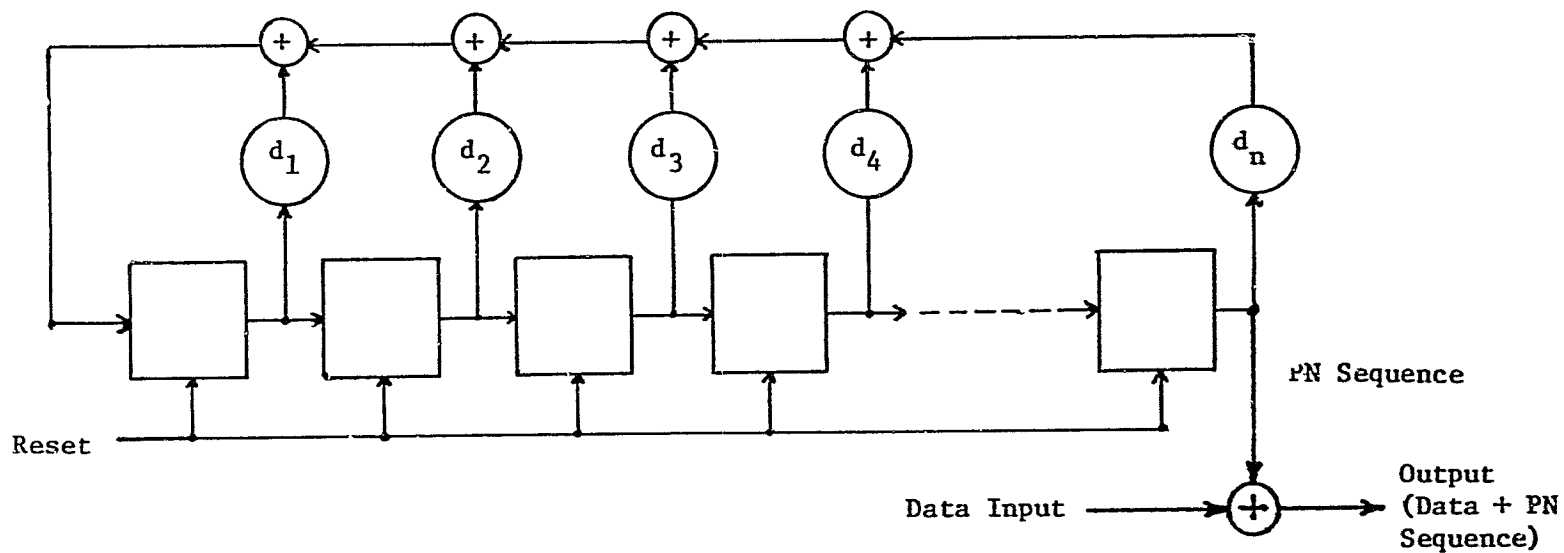


Figure 4.1 PN Cover Sequence. The d_i ($i = 1, 2, \dots, n$) are either one or zero, depending on the tap polynomial, $h(x)$, where $h(x)$ is a primitive polynomial over the field $GF(2^n)$ and $h(x) = 1 + d_1x + d_2x^2 + d_3x^3 + \dots + d_nx^n$.

distribution of ones and zeros is well defined and always the same. There are exactly $2^{n-(p+2)}$ runs of length 'p' for both ones and zeros in every maximal sequence where 'p' is any positive integer less than 'n', including zero. However, the relative positions of their runs vary from ML sequence to ML sequence depending on the tap polynomial and the method of connection. The main properties of ML sequences are listed below.

1. The number of ones in a sequence equals the number of zeros within one bit.
2. The statistical distribution of ones and zeros is well defined and always the same. Relative positions of their runs vary from code sequence to code sequence, depending on the tap polynomial and the method of connection, but the number of each run length does not.
3. Autocorrelation of a maximal linear code sequence is such that for all values of phase shift the correlation value is -1, except for the 0 to 1 bit phase shift area, in which correlation varies linearly from the -1 value to $2^n - 1$ (the sequence length).
4. A modulo-2 addition of a maximal linear code with a phase shifted replica of itself results in another replica with a phase shift different from either of the originals.
5. Every possible state, or m-tuple, of a given n-stage generator exists at some time during the generation of a complete code cycle. Each state exists for only one clock pulse. The exception is that the all-zeros state does not normally occur and cannot be allowed. (For additional information concerning ML sequences, see Reference 13, pp. 53-72 and Reference 14.)

The particular ML sequence, the reasons for choosing it, and a statistical statement of the probability of not incurring sufficient transitions are discussed in the following paragraphs.

4.B THE PARTICULAR PN SEQUENCE FOR THE COVER SEQUENCE GENERATOR (CSG)

It is a common engineering practice to select a ML sequence whose length is at least equal to the number of bits between synch words, since the sync pattern is better left alone. The SL general user format contains 3040 bits between each 32 bit sync word (28 bit sync pattern and a 4 bit ID word), therefore the sequence generated by the CSG should have a length equal to or greater than 3040. Although a 12^{th} degree polynomial yields a ML sequence length of 4095 bits and is the smallest sequence that could be used, it is not a Mersenne prime sequence and is therefore susceptible to interperiodicity. For this reason a 13^{th} degree polynomial was chosen. It will generate a $2^{13}-1 = 8191$ bit sequence before repeating itself; its composition will vary depending upon the tap polynomial used. A 13 stage PN generator is a Mersenne prime generator. Various tap polynomials are available for use and some are listed as follows:

$$g(x) = 1 + x + x^3 + x^4 + x^{13}$$

$$g(x) = 1 + x^4 + x^5 + x^7 + x^9 + x^{10} + x^{13}$$

$$g(x) = 1 + x + x^4 + x^7 + x^8 + x^{11} + x^{13}$$

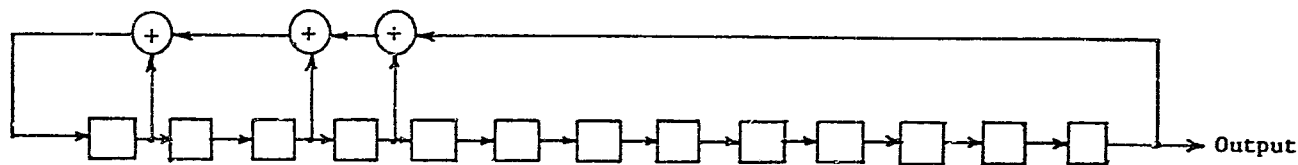
etc.

The first polynomial listed yields the fewest number of connections which would be desirable if a shift register implementation was used for producing the code.

Since the sequence generated is 8191 digits long and only 3040 digits exist between the frame synchronization patterns, a truncation of the sequence is desirable.

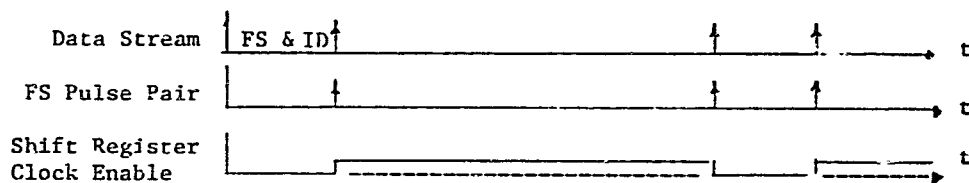
Deciding upon the particular 3040 bit piece of the sequence is dependent upon the structure of the sequence. It is highly desirable to avoid long strings of alternating 1's and 0's due to the very likely prospect of these strings occurring in the data. Factors which enter into the sequence structure are the initial condition (or contents) of the PN sequence shift register and the tap polynomial.

Figures 4.2 and 4.3 illustrate the methods of modifying the data stream emanating from the HRM. Figure 4-2 illustrates the shift



Shift Register for $g(X) = 1 + X + X^3 + X^4 + X^{13}$

(a)



(b)

OPERATION: For Frame Synch and ID (32 bits) the Shift Register output is "0". Thus Frame Synch and ID pass to transmitter unaltered. For the next 3040 bits the PN sequence (a specific 3040 bit portion of the 8191 total) is added to data thereby producing the required transition density in the transmitted bit stream

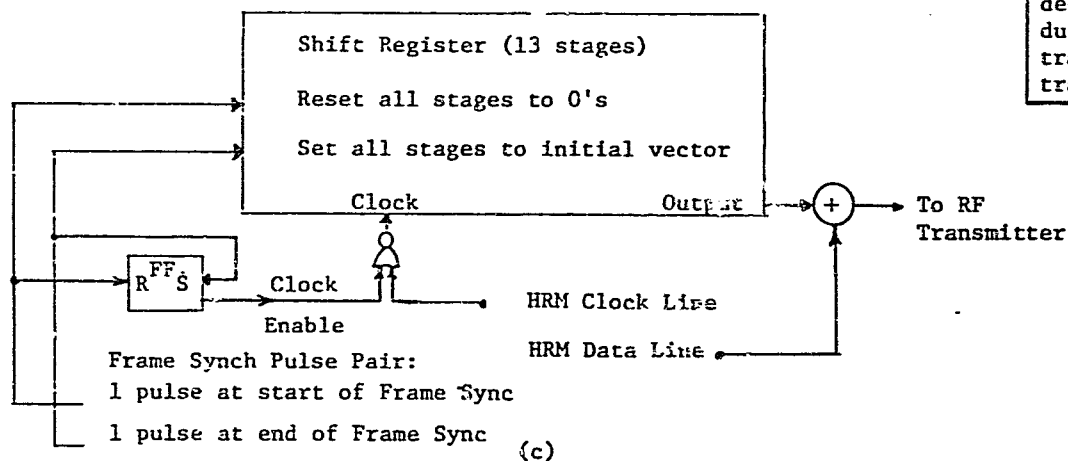


Figure 4.2 CSG Encoder Using Shift Register Implementation

ORIGINAL PAGE IS
OF POOR QUALITY

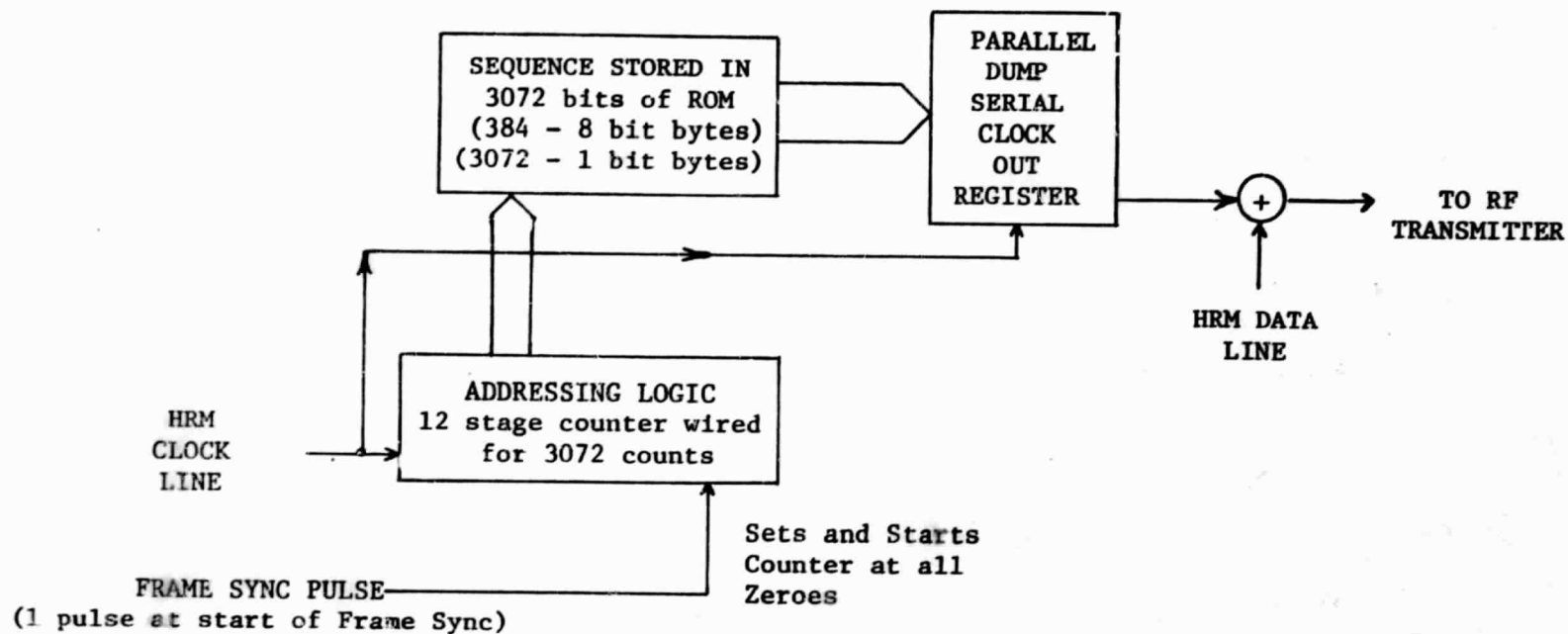


Figure 4.3 HDBT Encoder Using ROM Implementation

ORIGINAL PAGE IS
OF POOR QUALITY

register type of encoder for HDBT and Figure 4.3 illustrates the ROM type of encoder for HDBT. In either case the same sequence will be utilized. It should be noted that the frame sync and ID portion are not altered.

The sequence will have the run length distributions, shown in Table 4.1, for all tap polynomials and initial start vectors (the arrangement of these various runs will vary of course or there would be no difference in the sequences).

In general there are $2^{n-(p+2)}$ runs of length p for both ones and zeros in every maximal sequence, except that there is only one run of length n (ones) and one run of length $n-1$ (zeros).

A computer program based on an algorithm presented by Robert Gold in Reference 14 was used to generate the complete sequence of 8191 bits. This sequence was examined and the particular portion of 3040 bits was selected. This sequence is shown in Table 4.2. The zero-one distribution for this truncated 13 stage PN sequence is illustrated in Table 4.3. An examination of Table 4.3 reveals that this truncated sequence maintains the properties of a ML sequence. Although they are not perfectly retained, it is very close. Since the SL data stream is expected to contain long runs of alternate ones and zeros, the truncated sequence must be examined for these also. Table 4.4 lists the number and lengths of all alternating runs contained in the truncated sequences.

4.C PROBABILITIES ASSOCIATED WITH THE SEQUENCE

Now let us investigate some probabilities of not achieving the transition density requirements.

In order for failure to achieve a transition in 64 bits to occur it must have a data sequence that exactly matches the PN sequence for 64 bits. Since the PN sequence is statistically independent of the data sequence in bit by bit as well as string by string fashion we have

$$\begin{aligned} &\text{Prob (of more than 63 bits with no transition)} \\ &= (.5)^{64} = 5.4210108 \times 10^{-20} \end{aligned}$$

The requirement that 64 transition in 512 bits occur may be thought of from the following viewpoint:

TABLE 4.1 ZERO-ONE DISTRIBUTION FOR $n = 13$
MAXIMAL LENGTH PN SEQUENCE

TOTAL NUMBER OF ONES	4096
TOTAL NUMBER OF ZEROS	4095
TOTAL BITS	8191

<u>RUN LENGTH</u>	<u>NUMBER OF RUNS CONSECUTIVE ONES</u>	<u>NUMBER OF RUNS CONSECUTIVE ZEROS</u>	<u>NUMBER OF BITS INCLUDED</u>
13	1	0	13
12	0	1	12
11	1	1	22
10	2	2	40
9	4	4	72
8	8	8	128
7	16	16	224
6	32	32	384
5	64	64	640
4	128	128	1024
3	256	256	1536
2	512	512	2048
1	1024	1024	2048
		TOTAL BITS	8191

TABLE 4.2 TRUNCATED SEQUENCE FOR THE CSG

1010000101111001	0010100001111110	0100101111011000
100001000000111	0010010010111001	0001101000110110
0101010001010110	0000001101001001	0011010100101011
1011100101000001	0101100011101110	1010111110101100
0010110000111001	0110110011101111	1011001111010011
0111101110011110	1011111101001111	0100010011001010
1001011010110001	0000011000110110	1111111011111010
0001100110101010	0011000001101100	1001000001001110
1101100010110010	1101110101001101	0001000110000000
1110101010101111	1001010011001110	1110011101001100
0101100010101001	1011000001111001	1100010111001101
1000010011010010	0111010110101000	0011000001110000
1110000100100001	1100100011101001	0101000001111100
1001000010101101	0101001111001010	0101111110010001
0111100000001100	1110001111000010	0101111111100000
1011110110110000	1010001010101100	0110101100100110
1011100100000110	0100011100111011	0100011001011000
1100010000000100	1010101010010100	1111100000111101
0010010111110000	0011001101101100	1000001000000111
00000000000101010	1010101100000110	1011100011000001
0101000111001010	0100001111100000	0001011100011100
1011100001001111	0110111001100001	1001000000001001
1100011101100111	0011010010111001	1111100110111101
0010110011010100	1101111100100000	1011100100100010
1101010010001010	0111011010101100	0101001101010110
00101001111100010	0011001111001001	0111100000010000
1001001010101101	0100111110111011	0011000010000001
0010001110000001	1101001001000011	0111001001011001
1101100001111011	1111110100110011	0010101110011101
1101001011101001	1101111001000011	0011101101111110
1011101000011010	0010010000011101	1000111010011011
0011111101111110	1000001011111000	1111101000111101
0011100110000001	0101110001111100	1101100110001010
0011000110101011	1000011010111111	1101110100001010
0000111001010101	0000001101011011	0111110011110011
0010000100110111	0001001101000111	1110111000010111
0100100111101110	0100001000011111	1110001100110011
1001110101000000	1010011100011010	100000010001001
0010001111110000	0001011111111111	0011001100110111
1110110010111101	1111100110000101	1100111000001010
1111111110010111	1010001101000001	1101011011010001
0000010001111111	1100010111101001	0001011101100011
0100010101000100	1010011100100010	0110001001010111
0000001101000111	0000110110011100	0011000101101100
1001111001110110	0110111100111010	1111000010001011

TABLE 4.2 TRUNCATED SEQUENCE FOR THE DSG (Continued)

1000100101000000	0111110001110011	0010011000101011
0100100010000011	1111100011110100	0000010110001110
0000100101110001	1011101001000001	0100101010100111
0111011100110100	0101101001110010	1100010110101110
0000100010110110	1010110010110000	0100111100011111
1010010000101100	0100100010101001	0101001111110010
1011110101000111	0101100010111011	1111100110100001
0101110110111011	1100111101111011	0011010000010011
0101010110100111	1100111111010001	1001100000000100
1011011010100010	1000100011111000	1001011110001001
1110101010101000	0110100111000101	0010111100000111
1010101011110111	0111010001011001	1111110011101000
0100110000000011	1100011100001010	1000000001101010
1010100010001000	1001010100100011	0101101010111011
1110010111010000	0011001010101011	1001010011110110
0000010110010010	0111100000001110	1010101000011010
1100011110010110	1111011000011001	1110001100010111
0000111011110001	1001011111110110	0001011111011011
1010000010000110		

TABLE 4.3

Truncated

13 STAGE PN SEQUENCE PROPERTIES

$$g(X) = 1 + X^9 + X^{10} + X^{12} + X^{13} \quad (\text{MSRG})$$

Total Number of Ones - 1504

Total Number of Zeroes - 1536

TOTAL DIGITS 3040*

<u>RUN LENGTH</u>	<u>NUMBER OF RUNS CONSECUTIVE ONES</u>	<u>NUMBER OF RUNS CONSECUTIVE ZEROES</u>	<u>NUMBER OF BITS INCLUDED</u>
11	1	0	11
10	0	1	10
9	2	0	18
8	4	4	64
7	5	10	105
6	14	12	156
5	20	30	250
4	45	41	344
3	101	93	582
2	173	193	732
1	395	373	768
TOTAL BITS			<u>3040</u>

* Maximum Run of Alternate One/Zero is 14

TABLE 4.4

DISTRIBUTION OF ALTERNATE ONES/ZEROS FOR
THE TRUNCATED 13 STAGE PN SEQUENCE

RUN LENGTH (BITS)	NUMBER OF RUNS
14	1
12	1
11	2
10	2
9	4
8	4
7	16
6	21
5	35
4	102

The PN sequence is random in nature and contains approximately equal numbers of ones and zeros with a corresponding large number of bit transitions. In fact from the run length table one may observe that approximately 4096 transitions between various run lengths will occur. This amounts to roughly a 50% (or 1 transition per 2 bits) transition density. Thus, for less than 64 transitions to occur in a total of 512 bits we must have the data match the 512 bit random sequence in all bit positions except for 63 bits or 62 bits, or 61 bits, etc.

$$\begin{aligned}
 P(\leq 64/512) &= \sum_{k=0}^{63} \binom{512}{k} (.5)^{512} & (4.1) \\
 &= 3.946^+ \times 10^{-71}
 \end{aligned}$$

Thus, we see that the dominant factor is the probability of less than at least 1 transition in 64 bits which is $\sim 5.42 \times 10^{-20}$.

Of course there are 448 possible chances for a 64 bit string to have no transitions in 512 bits.

Thus, the transition density requirements should be met with at least a failure probability of no more than

$$\sim 2.434 \times 10^{-17} .$$

A computer simulation program was developed which tests the truncated CSG Sequence for achieving the high bit density transitions by modulo-2 addition with the data stream. This program is explained and listed in the Appendix. Since little is known about the SL data, the program generates a sequence of random numbers to represent the SL data stream. Several runs were made using different seed numbers to produce different random sequences. These random sequences were also truncated to various lengths and repeated to simulate periodic data sequences. The computer results indicate that the output sequence of the CSG will have a transition density of approximately 50%, an average of one transition every two bits. Although the computer simulation was not exhaustive, it is sufficient to substantiate the theoretical probability of meeting the required bit transition density of the SL 2 Mbit return link.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

In conclusion only the PN Cover Sequence of the six possible techniques examined is capable of meeting or exceeding the System Constraints placed on the HBTD encoder (See Table 5.1). Since the PN Cover Sequence is NRZ-L and employs the HRM clock, it will not alter the present bandwidth or data rate. The probability of not meeting or exceeding the required bit transition density is at most 2.434×10^{-17} . Since the PN Cover Sequence is independent of the data stream, it will not propagate channel errors, a single input error yields a single output error. Thus the PN Cover Sequence is compatible with the existing BCH code. The above mentioned are three primary constraints listed in Table 2.1. The fourth constraint deals with the means use to implement the CSG and will vary depending on whether shift register or ROM techniques are employed. This constraint is also effected by the Secondary Constraints in Table 2.2. However the circuitry required to meet the secondary constraints is basically the same for all six techniques and since the truncated PN Sequence is one of the least complicated to implement, the fourth constraint presents no major problem.

Since the actual implementation of the CSG is beyond the scope of this contract, it was not covered in the preceding material. However, a specifications document concerning the PN Cover Sequence Generator Encoder/Decoder was constructed and submitted with the April 1981 monthly progress report. A copy of this specifications document is included in the Appendix. It contains diagrams for the purpose of enhancing the concept of the CSG encoder and decoder. These figures illustrate the functional properties desired for both the CSG encoder and decoder which will enable them to meet the secondary constraints of Table 2.2.

A second method of implementation was also put forth in the November 1980 Monthly Report for meeting the secondary constraints. This method differs in that the 4 ID Bits are used to designate whether the CSG encoder had been activated or by passed. This modification

varies only slightly from the original truncated sequence, 4 additional bits are added increasing the sequence length to 3044 bits. This would not effect any of the first three constraints of Table 2.1 and would require moderate changes in the implementation. A copy of the November 1980 Monthly Report is included in the Appendix for convenience.

TABLE 5.1 COMPARISON OF CSG WITH SYSTEM CONSTRAINTS

- . NO CHANGE IN RF BANDWIDTH
- . NO CHANGE IN DATA BANDWIDTH
- . PROBABILITY OF NOT MEETING TRANSITION REQUIREMENTS
 - 1 TRANSITION IN 64 BITS - 5.421×10^{-20}
 - 64 TRANSITION IN 512 BITS - $3.946^+ \times 10^{-71}$
 - 1 TRANSITION IN ANY 64 BITS GROUP FOR 513 BITS - 2.434×10^{-17}
- . NO ERROR PROPAGATION
- . HARDWARE IMPLEMENTATION
 - AIRBORNE: SIMPLE (FRAME SYNC PULSES AND CLOCK PROVIDED BY HRM)
 - GROUND: MODERATE (REQUIRES FRAME SYNC DETECTION FOR NORMAL AND INVERTED AND MUST DETERMINE DATA RATE.)

CHAPTER 6

References

1. Baillavoine, J. "High Rate Acquisition Assembly; Equipment Specification," MATRA ESPACE, Doc. # EQ. MA. 0229, VL'LIZY, France, 1977.
2. _____ "System Description and Operation; High-Rate Demultiplexer System Operations and Maintenance Manual," Martin Marietta, Denver, Colorado, Volume 1, MCR-80-1328, 1980.
3. Teasdale, W. E. and Lai-lum Lo. "Space Shuttle Payload Data Handling and Communication Description and Performance Document," Revision A prepared by NASA, Johnson Space Center, Houston, Texas, JSC-14241. 1980.
4. Tracking and Data Relay Satellite System (TDRSS) User's Guide, Revision 4, prepared by Goddard Space Flight Center, Greenbelt, MD., 1980.
5. Memo from Networks Directorate, Network Engineering Division, Goddard Space Flight Center, Greenbelt, MD to Data Systems Laboratory, Mr. Russell D. Coffey, Marshall Space Flight Center, "Tracking and Data Relay Satellite System (TDRSS) Shuttle Return Link Channel 2 Bit Synchronization Performance Under Low Data Transition Density Conditions,".
6. Simon, M. K. and J. G. Smith. "Alternate Symbol Inversion for Improved Symbol Synchronization in Convolutionally Coded System," IEEE Trans. Commun., Vol. COM-28, Feb. 1980.
7. Cartier, D. E. and D. E. McConeell, "Space Telescope High Data Rate System Study," Magnavox Government and Industrial Elec. Co., Silver Spring, MD, Contract No. NAS5-24225, prepared for Goddard Space Flight Center, Greenbelt, MD, July 1979.
8. Baumert, L. D. et.al., "Symbol Synchronization in Convolutionally Coded System," IEEE Trans. Inform. Theory, Vol. IT-25, pp 362-365, May 1979.
9. Savage, J. E., "Some Simple Self-Synchronizing Digital Data Scramblers", Bell System Technical Journal, Vol. 54, No. 3, Mar. 1975, pp 569-593.
10. Gitlin, R. D. and J. F. Hayes. "Timing Recovery and Scramblers in Data Transmission," B.S.T.J., Vol. 54, No. 3 (Mar. 1975); pp 569-593. B, E, (Parallel Scramblers) 2b

11. Spilker, J. J., Jr. Digital Communication by Satellite, McGraw-Hill, 19 , pp. 473-491.
12. Muller, Horst. "Bit Sequence Independence Through Scramblers in Digital Communication Systems," Nachrichtentechnische Zeitschrift, Vol. 27, Dec. 1974, pp. 476-479.
13. Dixon, R. C. Spread Spectrum Systems, New York, N. Y., Wiley and Sons, 1976.
14. . "Properties of Linear Binary Encoding Sequences," Short Course prepared by Robert Gold Associates, Los Angeles, CA., Sept. 1978.
15. Lindsey, W. C. and M. K. Simon, Telecommunication Systems Engineering, Englewood Cliffs, NJ, Prentice-Hall, 1973.
16. Ziemer, R. E. and W. H. Tranter, Principles of Communications Systems, Modulation, and Noise, Houghton Mifflin Co., 1976.

CHAPTER 7

APPENDICES

C-2

APPENDIX A

COMPUTER SIMULATION PROGRAM

A computer simulation program was developed which tests the PN-Sequence for achieving the high bit density transitions by modulo-two addition with the data stream.

The computer program will generate several different PN-Sequences for given generating polynomials (represented by IX) with given initial conditions (represented by IG). Both IX and IG are written in octal representation. The program generates, via subroutines, a sequence of random numbers which simulates the input data stream to the Cover Sequence Generator (CSG) Encoder. Any specified portion of this random sequence can be used to simulate a periodic input sequence. Thus the program can generate periodic sequences of varying lengths and transition densities to simulate the input data stream to the CSG Encoder. The modulo-two sum of the PN-Sequence and the simulated input data is calculated. This modulo-two sum represents the output data stream of the CSG Encoder. All three sequences are printed and the total number of transitions for each is determined and printed.

The program will also determine for the output sequence the run lengths for all bits of the same value and the order of their occurrence. This information is stored and printed in two groups, one for the all zeros case and the other for the all ones case. Examination of these two groups permits a determination of whether the output sequence satisfies the required transition density.

The status of the requirement of one transition every sixty-four bits is determined by simply checking the number of bits per run in each group. The second requirement of sixty-four transition within five-hundred-twelve bits is slightly more complicated to evaluate. An average of one transition every eight bits will satisfy this requirement. Thus by locating all runs of length eight or greater and examining the runs prior to and following each of these, a determination concerning this requirement can be made. Note the output sequence alternates between the two groups. The question of which group to start with, zeros or ones, depends upon the value of the first output bit ('XOR OF THE

GENERATED BITS'). If the bit is a one start with the "LENGTHS OF GROUPS OF ONES" and if the first output bit is a zero start with the "LENGTHS OF GROUPS OF ZEROS".

ASYM PRINTS,,FORMS

BFTN,SIC
FTN 9R1

*09/08/81-12:15

```

1. DIMENSION ITRZCT(3,20),IK(3100),JC(1500),KC(1500)
2. DIMENSION IG(20),IX(20),K(3,3100)
3. FORMAT(1H1)
4. 2
5. FORMAT(3(/))
6. C*** ITRZCT(.,.) IS THE TRANSITION ARRAY
7. C*** IK(.,) IS ARRAY OF NUMBERS WHOSE SORTING OF ONES AND ZEROS IS
8. C*** DESIRED.
9. C*** JC(.,) IS THE VECTOR CONTAINING # OF ONES IN EACH GROUPS
10. C*** KC(.,) IS THE VECTOR CONTAINING # OF ZEROS IN EACH GROUP
11. C*** IG(.,) IS INITIAL CONDITION FOR GENERATING THE PN-SEQUENCE
12. C*** IX(.,) IS THE GIVEN INFORMATION TO GENERATE THE PN-SEQUENCE
13. C*** K(1,.) IS THE PN-SEQUENCE ARRAY
14. C*** K(2,.) IS THE INPUT SEQUENCE
15. C*** K(3,.) IS THE ARRAY CONTAINING MOD-2 SUM OF PN-SEQ AND THE INPUT
16. C*** DO LOOP KCC IS FOR THE NUMBER OF REPETITIONS IF IT IS DESIRED
17. C*** TO HAVE REPETITIVE SEQUENCES WITH DIFFERENT LENGTH TO BE REPEATED
18. C*** N IS THE TOTAL NUMBER OF TIMES THAT THE PN-SEQUENCE ,INPUT SEQUENCE
19. C*** AND MOD-2 SUM SEQUENCES ARE RUN IN EACH KCC RUN
20. C*** M IS TOTAL NUMBER OF POINTS(-ITS) IN EACH ARRAY
21. C*** NLN IS THE LENGTH OF SEQUENCE TO BE REPEATED
22. C*** NPS IS THE BIT NUMBER IN EVERY INPUT SEG TO MAKE THE PERIODIC
23. C*** SEQUENCE.
24. C***
25. C*** READ TOTAL NUMBER OF DESIRED SEQUENCES AND TOTAL NUMBER
26. C*** OF BITS IN EACH SEQUENCE
27. C***
28. DO 1000 KCC=1,1
29. K1=KCC-1
30. KE=2**K1
31. NLN=1(24/KE
32. NPS=130
33. READ(5,10) N,M
34. 10 FORMAT(2I5)
35. NP=N
36. M2=M/2
37. WRITE(6,1)
38. WRITE(6,15) NPS,NLN
39. 15 FORMAT(10X,20HSTARTING POINT AT ,110,10X,31HLENGTH OF THE SEQUEN
40. *CE EQUAL ,110,/)
41. C***
42. C*** WRITE OUT TOTAL NUMBER OF SEQUENCES AND THEIR BITS
43. C***
44. WRITE(6,20) N,M
45. 20 FORMAT(1H0,15X,2HN=,15,10X,2HM=,15)
46. C***
47. C*** READ AND WRITE INPUT NUMBERS TO GENERATE THE PN-SEQUENCE
48. C***
49. DO 30 I=1,N
50. READ(5,40) IG(I),IX(I)
51. CONTINUE
52. 30 FORMAT(205)
53. DO 50 I=1,N
54. WRITE(6,60) IG(I),IX(I)
55. CONTINUE
56. 50 FORMAT(15X,3HIG=,05,15X,3HIX=,05)
57. C***
58. C*** IN THE FOLLOWING LOOP RANDOM NUMBER SEQUENCES ARE GENERATED
59. C*** THEIR XOR WITH PN-SEQUENCES ARE FOUND AND ALL SEQUENCES
60. C*** ARE PRINTED OUT. TOTAL NUMBER OF TRANSITIONS FOR EACH
61. C*** SEQUENCE IS FOUND AND ARE PUT IN ARRAY ITRZCT FOR FUTURE
62. C*** USE AND PRINTOUT
63. C***
64. NC=1
65. CALL PNSEQ(IG,IX,M,NC,K)
66. IPOW=1
67. 80 WRITE(6,80)
68. FORMAT('1',////,10X,'*** BITS GENERATED BY THE FIRST METHODE ***
69. + ')
70. CALL WRITER(IPOW,K)
71. CALL TRANS(1,K,ITRZCT,NC,M)
72. DO 110 NC=1,N
73. CALL BTRAND(K,M,NC)
74. CALL RNCYCL(NPS,NLN,K,M)

```

```

74.      CALL AMOD2(K,M)
75.      IROW=2
76.      WRITE(6,90)
77.      90      FORMAT('1',/////,10X,'*** BITS GENERATED USING RANDOM NUMBERS ***'
78.      +)
79.      CALL WRITER(IROW,K)
80.      IROW=3
81.      WRITE(6,100)
82.      100     FORMAT('1',/////,10X,'*** XOR OF THE GENERATED BITS ***')
83.      CALL WRITER(IROW,K)
84.      CALL TRANS(2,K,ITRZCT,NC,M)
85.      CALL TRANS(1,K,ITRZCT,NC,M)
86.      110     CONTINUE
87.      WRITE(6,1)
88.      WRITE(6,120)
89.      120     FORMAT(10X,22HPN-SEQUENCE TRANSITION,10X,26HRANDON SEQUENCE TRANSI
90.      *TION,10X,26HOUTPUT SEQUENCE TRANSITION)
91.      DO 130 I=1,N
92.      ITRZCT(1,I)=ITRZCT(1,I)
93.      WRITE(6,140) ITRZCT(1,I),ITRZCT(2,I),ITRZCT(3,I)
94.      130     CONTINUE
95.      140     FORMAT(1H0,17X,15,31X,15,31X,15)
96.      DO 150 I=1,M
97.      IK(I)=K(3,I)
98.      CALL SORT(IK,M,M2,JC,KC,LC,MC)
99.      150     CONTINUE
100.      NMX=MAX0(LC,MC)
101.      NMX=NMX-1
102.      WRITE(6,1)
103.      WRITE(6,170)
104.      170     FORMAT(10X,35HTHESE ARE LENGTHS OF GROUPS OF ONES///)
105.      WRITE(6,180) (JC(I),I=1,NMX)
106.      180     FORMAT(25(14,1H,))
107.      WRITE(6,2)
108.      WRITE(6,190)
109.      190     FORMAT(10X,35HTHESE ARE LENGTHS OF GROUPS OF ZEROS///)
110.      WRITE(6,200) (KC(I),I=1,NMX)
111.      200     FORMAT(25(14,1H,))
112.      1000   CONTINUE
113.      WRITE(6,1)
114.      STOP
115.      END

```

```

116.      SUBROUTINE PNSEQ(IG,IX,M,NC,K)
117.      DIMENSION IG(20),IX(20),K(3,3100)
118.      C***
119.      C*** THE PNSEQ SUBROUTINE WILL GENERATE A SET OF PSEUDO-RANDOM
120.      C*** BITS. IN THIS ROUTINE SUBFUNCTION BITS FROM THE LIBRARY
121.      C*** OF UNIVAC 1100 HAS BEEN UTILIZED
122.      C***
123.      M6=M
124.      NTP=NC
125.      IHOLD=IG(NTP)
126.      I=0
127.      10      IG(NTP)=IG(NTP)*2
128.      IG(NTP)=IG(NTP)/2
129.      I=I+1
130.      IF(I.GT.M6) GO TO 20
131.      K(1,I)=BITS(IG(NTP),36,1)
132.      IF((1.GT.1).AND.(IG(NTP).EQ.IHOLD)) GO TO 20
133.      IF(K(NTP,1).EQ.0) GO TO 10
134.      IG(NTP)=XOR(IG(NTP),IX(NTP))
135.      GO TO 10
136.      20      CONTINUE
137.      RETURN
138.      END

```

```

139.      SUBROUTINE WRITER(IROW,K)
140.      DIMENSION K(3,3100)
141.
142.      C*** THIS SUBROUTINE WRITES OUT A SET OF GENERATED RANDOM
143.      C*** NUMBERS
144.      C*** IROW=1 FOR PN-SEQUENCE
145.      C*** IROW=2 FOR INPUT SEQUENCE
146.      C*** IROW=3 FOR MOD-2 SUM
147.
148.      C***
149.      JLINE=0
150.      DO 10 IOUT=1,31
151.      NP=IOUT*100
152.      NT=((IOUT-1)*100)+1
153.      WRITE(6,20) (K(IROW,L),L=NT,NP)
154.      JLINE=JLINE+1
155.      IF(JLINE.LE.23) GO TO 10
156.      CONTINUE
157.      FORMAT(1H ,//17X,100(I1))
158.      RETURN
159.      END

```

```

159.      SUBROUTINE TRANS(JB,K,ITRZCT,NC,M)
160.      DIMENSION K(3,3100),KK(3100),ITRZCT(3,20)
161.
162.      C*** THIS SUBROUTINE FINDS THE TOTAL NUMBER OF TRANSITIONS
163.      C*** IN ANY SEQUENCE OF BINARY NUMBERS
164.      C*** ITRZCT(1,..) IS FOR PN SEQUENCE
165.      C*** ITRZCT(2,..) IS FOR INPUT SEQUENCE
166.      C*** ITRZCT(3,..) IS FOR MOD-2 SUM
167.
168.      C***
169.      ITZ=0
170.      DO 10 I=1,M
171.      KK(I)=K(JB,I)
172.      CONTINUE
173.      DO 20 I=2,M
174.      I1=I-1
175.      IF(KK(I).EQ.KK(I1)) GO TO 20
176.      ITZ=ITZ+1
177.      ITRZCT(JB,NC)=ITZ
178.      CONTINUE
179.      RETURN
180.      END

```

```

180.      SUBROUTINE BTRAND(K,M,NC)
181.      DIMENSION K(3,3100)
182.
183.      C*** THIS SUBROUTINE GENERATES A SET OF PSEUDO-RANDOM BITS
184.      C*** IN COOPERATION WITH SUBROUTINE RAND
185.      C***
186.      N=13573951373
187.      N=N/NC
188.      DO 10 J=1,M
189.      CALL RAND(R,N)
190.      R=R*2
191.      IBIT=R
192.      K(2,J)=IBIT
193.      CONTINUE
194.      RETURN
195.      END

```


ORIGINAL PAGE IS
OF POOR QUALITY

```

196.      SUBROUTINE RAND(R,N)
197.      C***
198.      C*** THIS SUBROUTINE GENERATES A SET OF RANDOM NUMBERS
199.      C***
200.      LK=315227
201.      N=N*LK
202.      RN=N
203.      R=RN/34359738337.
204.      R=ABS(R)
205.      RETURN
206.      END

```

```

207.      SUBROUTINE RNCYCL(NPS,NLN,K,M)
208.      DIMENSION K(3,3100),KOK(1024)
209.      C***
210.      C*** THIS SUBROUTINE WILL TAKE RANDOM SEQUENCE GENERATED BY
211.      C*** CTRAND AND STARTING AT POINT NPS IN THE SEQUENCE CHOOSES
212.      C*** NLN BITS OF THE SEQUENCE AND PUTS IT IN ARRAY KOK
213.      C*** THEN STARTS TO REWRITE THE ARRAY K WITH THE NCY+ OF NLN
214.      C*** BITS PERIODIC SEQUENCES. SO IN THIS OPERATION ORIGINAL
215.      C*** SEQUENCE K(2,I),I=1,2,...,M IS COMPLETELY LOST.
216.      C***
217.      NPML=NPS+NLN-1
218.      NCY=IFIX(M/NLN)
219.      NCY1=NCY+1
220.      NNCY=NCY*NLN
221.      DO 10 I=NPS,NPML
222.      I1=I-NPS+1
223.      KOK(I1)=K(2,I)
224.      10 CONTINUE
225.      DO 20 J=1,NCY1
226.      DO 30 I=1,NLN
227.      IJ=1+(J-1)*NLN
228.      K(2,IJ)=KOK(I1)
229.      IF(IJ.EQ.M) GO TO 40
230.      20 CONTINUE
231.      30 CONTINUE
232.      40 CONTINUE
233.      RETURN
234.      END

```

```

235.      SUBROUTINE AMOD2(K,M)
236.      DIMENSION K(3,3100)
237.      C***
238.      C*** THIS ROUTINE FINDS MOD-2 SUM OF ANY TWO BINARY NUMBER
239.      C*** IN PN AND INPUT SEQUENCE
240.      C***
241.      DO 10 I=1,M
242.      J=1
243.      K(3,I)=XOR(K(1,J),K(2,I))
244.      10 CONTINUE
245.      RETURN
246.      END

```

```

247.      SUBROUTINE SQPT(IA,NP,NP2,JC,KC,LC,MC)
248.      DIMENSION IA(NP),KC(NP2),JC(NP2)
249.      C***
250.      C*** THIS SUBROUTINE WILL SORT ONES AND ZEROS IN ANY BINARY SEQUENCE

```

ORIGINAL PAGE IS
OF POOR QUALITY

66

```

251. C*** BY STARTING FROM FIRST BIT AND COUNTING UNTIL DIFFERENT DIGIT SHOWS
252. C*** AND THEN FIND THE TOTAL NUMBER OF ONES AND ZEROS IN EACH
253. C*** GROUP, PUT THEM IN ARRAYS JC(FOR ONES) AND KC(FOR ZEROS)
254. C*** LC IS TOTAL NUMBER OF GROUPS OF ONES
255. C*** MC IS TOTAL NUMBER OF GROUPS OF ZEROS
256. C***
257. J=0
258. K=0
259. LC=1
260. MC=1
261. DO 1000 I=1,NP
262. IF(IA(I).EQ.0) GO TO 20
263. IF(K.EQ.0) GO TO 10
264. KC(MC)=K
265. MC=MC+1
266. K=0
267. 10 CONTINUE
268. J=J+1
269. IF(I.LT.NP) GO TO 1000
270. 20 CONTINUE
271. IF(J.EQ.0) GO TO 30
272. JC(LC)=J
273. LC=LC+1
274. J=0
275. IF(I.EQ.NP) GO TO 1000
276. 30 CONTINUE
277. K=K+1
278. IF(I.EQ.NP) KC(MC)=K
279. 1000 CONTINUE
280. RETURN
281. END

```

END FTH 718 IPANK 20260 DBANK

ENTERING USER PROGRAM

ORIGINAL PAGE IS
OF POOR QUALITY

67

STARTING POINT AT

130

LENGTH OF THE SEQUENCE EQUAL

1024

N= 1
IG=17767

M= 3040
IX=20033

*** BITS GENERATED BY THE FIRST METHODE ***

10100001011100100101000011111001001011101100010000100000011100100100101110010001101000101100101
010001010110000000110100100100101001001011100101000001010110001110111010101111010110000101100
001110010101001110111101100111010010111011100111010111101001111010001001100101010010110101
000100000110001101101111101111010000110010101010001100000110110010010000010011101101100101100
1101110101001101000100011000000110101010101111001010011001110110011000101100010110001011001
000001111001110001011100110110000100101001001110110101000001100000111000011001000
111010010101000001111001001000010101010101011100101001011110010001011100000001100111000111100
0010010111111100000101110110110000101000101110000011001000110011100111011
0100011001011000110001000000100101010100100111100000111010010010111100000011001101100100
00100000011100000000000101010101010110000011010110001100000010101110000000010111
0001110010111000010011110110011000011001000000010011100011101100111001101001011100111110011011
11010010110010101011111001000010111001001000101101001000101001110110101101001010011010110
00101001111000100011001110010011011100000010001001010101001111011101100110000100000010010
0011100000011101001001000011011100100101110110000111011111101001100110011011100110110010
1110100111011100100001100111011011111010111010000110100010010000011101100011101001101100111110111
1110100001011111000111101000111010011101100000101011000111100110110011000101000110101011
10000110111111101110100001010000111001010100000110101101111001111001111001100110001
001101000111111011100001011101010110100001000011111110001100110011100111001110011100111001
000110101000000100010010010001111110000001011111111100110011001101111101100110111110011000
01011100111000001010111111100101110100011010000011110110110100100000100011111110001011101001
0001011101100011010001010100010010001110001001001100010011011000000110100011100001101100111000011
000101101100100111100110110011011100001000101110001001010100000011110001110011001100110
0010101101001000100001111110001110100000010110001110000100111000110111010010000010100101010
01110110111001100100010110011101000101100111000001000101101010110010110000010011110001111

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

70

[illegible]

71.

ORIGINAL PAGE IS
OF POOR QUALITY

[illegible]

73

73

PN-SEQUENCE TRANSITION

1515

RANDOM SEQUENCE TRANSITION

1458

OUTPUT SEQUENCE TRANSITION

1481

ORIGINAL PAGE IS
OF POOR QUALITY

74

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

SPECIFICATION DOCUMENT FOR PN COVER SEQUENCE
GENERATOR ENCODER/DECODER1.0 Introduction

This Statement of Work (SOW) is to establish the requirements for the definition and fabrication of Cover Sequence Generator (CSG) encoders and decoders and associated cabling hardware for utilization in Spacelab.

The encoder shall be utilized in the interface between the Spacelab HRM 2Mbps output and the Orbiter KU-band signal processor to format the bit stream to ensure minimum bit transition density.

The decoder shall be utilized in the interface between ground station receivers and HRDM(s) to restore the reformatted data to the non-encoded configuration and to resolve output bit ambiguity.

2.0 Scope

The contractor shall provide the necessary effort to define and fabricate:

(a) CSG encoders and associated cabling and hardware both for the Spacelab module and pallet modules.

(b) CSG decoders for implementation and integration into:

1. ATE at KSC
2. SLDPF at GSFC
3. POCC at JSC

The contractor shall provide the necessary effort to support the High Data Link Test scheduled at JSC - April 1982 by having available a CSG encoder/decoder qualification unit set and providing engineering and test support.

The contractor shall prepare and maintain a schedule plan which implements this SOW.

3.0 Deliverables

- . One (1) Set of Documentation
- . One (1) CSG E/D Qualification Set
- . Two (2) CSG Encoders
- . Three (3) CSG Decoders
- . Associated Hardware and Cabling

4.0 Specifications

The units shall be manufactured in accordance with Spacelab flight and GSE requirements. The quality program shall be in accordance with NHB 5300.4(1C). Functional characteristics shall be in accordance with Appendix B.1.

5.0 Acceptance

Acceptance tests will be performed at the contractor plant. The acceptance data package will be provided to NASA for review and approval. The review by NASA will be completed with 30 days of submittal.

6.0 Reviews

Design and development reviews shall include but not be limited to the following:

- PRR
- PDR
- CDR
- Acceptance Review (AR)

The contractor will present additional reviews as required.

APPENDIX B-1

COVER SEQUENCE GENERATOR ENCODER/DECODER
PRELIMINARY SPECIFICATION1.0 Scope

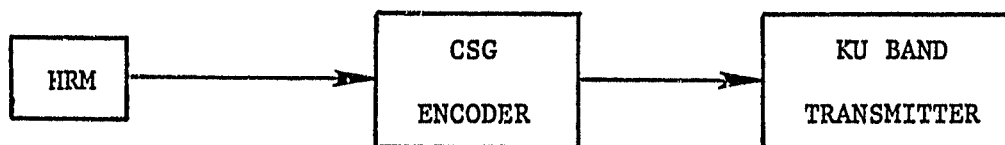
This appendix defines the preliminary requirements for design of the CSG Encoder/Decoder elements and will be used as the basis for development of the part 1 CEI Detail Specification (CM-04).

1.1 Purpose and Description

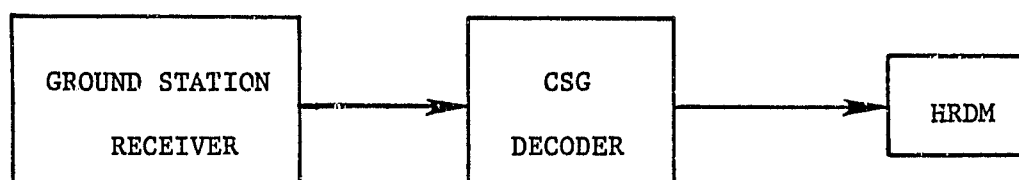
The Cover Sequence Generator Encoder serves the purpose of reformatting the Spacelab HRM 2MBPS serial bit stream in order to satisfy data handling constraints for minimum bit transitions density. The CSG Encoder will be physically located on board the Spacelab vehicle and will be functionally located between the Spacelab HRM 2 MBPS output and the Orbiter KU Band Signal Processor (KUSP).

A complementary decoder shall be utilized to restore the reformatted data to the non-encoded configuration. The decoder shall be functionally located in the interface between the ground station receiver and the HRDM. Figure B.1 illustrates the functional location of the encoder and decoder units. Figure B.2 illustrates the portion of the data stream to be reformatted and the portion of the data stream which is not to be reformatted. Note that use of the frame synchronization pulse generated by the HRM is necessary to turn the CSG Encoder on and off at the proper time. Note further that either the synthesis of the frame synchronization pulse or use of such a pulse from the HRDM will be necessary in the decoder. Furthermore, because of possible phase inversion of the data stream (including the frame synchronization word), the synthesis of the frame synchronization pulse if necessary will require a frame synchronization pattern or the inverted frame synchronization pattern. If however, the HRDM supplies the frame synchronization pulse it will take the possibility of inverted data into account.

The CSG encoder/decoder consists basically of a PN sequence which is derived using a 13th order polynomial. The sequence so generated is truncated to 3040 bits to match the data stream between frame synchronization pulses.

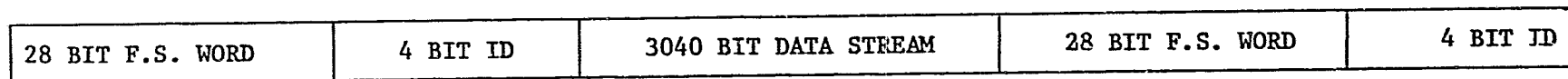


a) CSG Encoder Functional Location



b) CSC Decoder Functional Location

Figure B.1



This Portion has PN Sequence
Modulo Added

Figure B.2 HRM Data Stream

A printout of the desired 3040 bit sequence will be furnished to the contractor by the contracting agency. Figures B.3 and B.4 illustrate the functional properties desired for both the CSG encoder and decoder. These diagrams are provided for the purpose of enhancing the communication of the concept of the CSG encoder and decoder.

IN NO WAY ARE THE FIGURES B.3 AND B.4 INTENDED OR SHOULD BE CONSTRUED TO BE RECOMMENDED DESIGN APPROACHES.

The basic system requirements are:

For the Cover Sequence Generator (CSG) Encoder

1. Data input shall be NRZ-L from the HRM formatter varying discretely at a rate between 125 KBPS to 2 MBPS. The encoder shall utilize a frame synchronization pulse from the HRM to turn the PN encoder on and off.

2. Only HRM 2 MBPS data line (Data rates vary from 125 KBPS to 2 MBPS) formatted data will be CSG encoded.

For the Cover Sequence Generator (CSG) Decoder

1. If no HRM frame synchronization (nominal 2 MBPS data rate) word occurs in the data stream from the bit synchronizer the data shall be passed on unaltered.

2. For HRM 2 MBPS formatted science data with proper frame synchronization word the data outputs phase will be unaltered and the cover sequence removed.

In Figure B.4 the Timing Gate and the Time Coincidence Gate serve the purpose of determining that the clock and frame sync are indeed a 2 MBPS rate rather than a higher rate. The hold portion of the Time Coincidence Gate is to activate the cover sequence generator for a major portion of a frame.

It should be noted that the frame synchronization detector portion of the CSG decoder must contain capability for and follow the same frame synchronization search, acquire and maintenance mode protocol as that specified in the HRDM specifications.

These following statements describe the basic HRDM frame synchronization operation and frame synchronization pulse location.

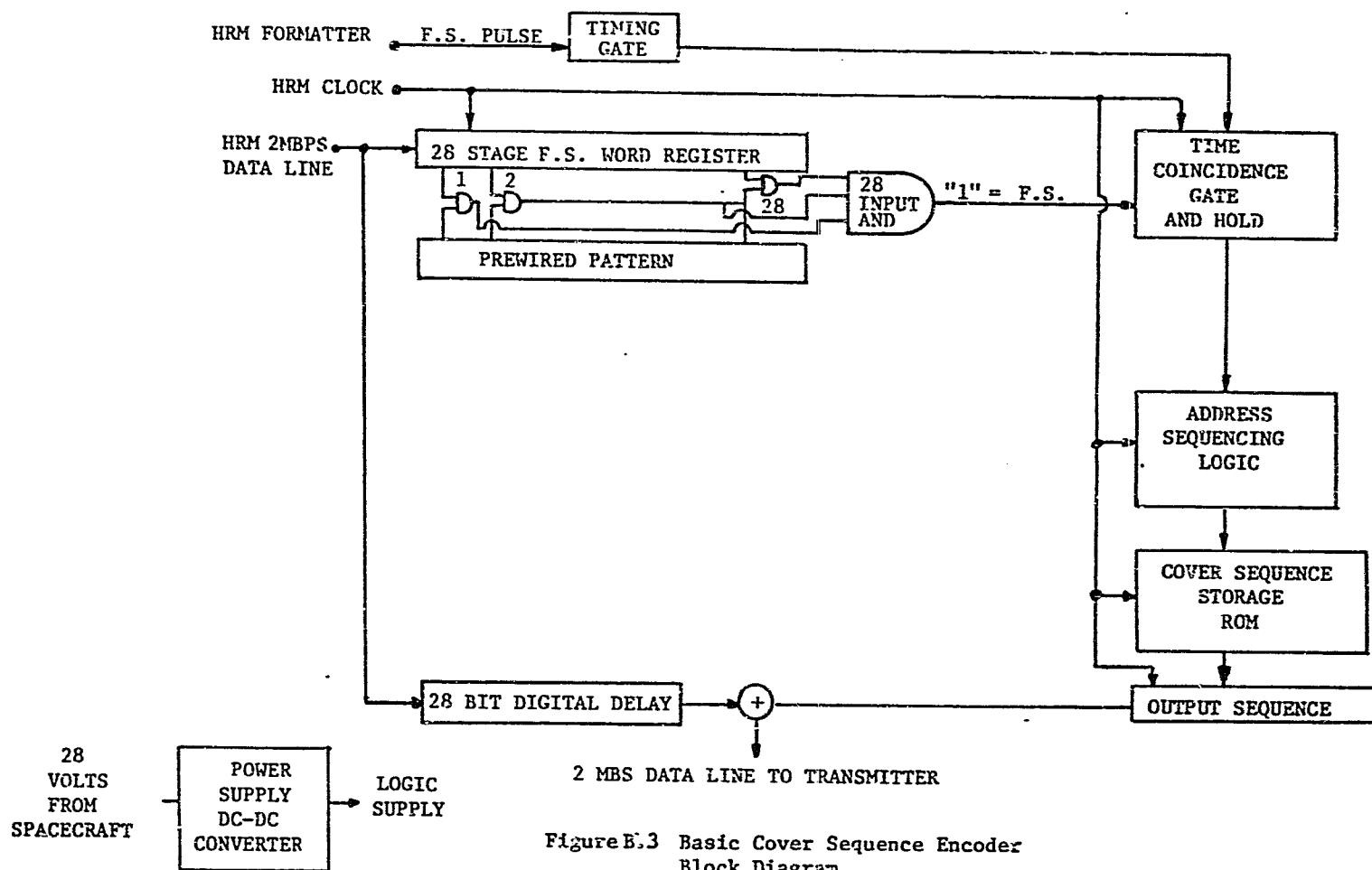


Figure B.3 Basic Cover Sequence Encoder
Block Diagram

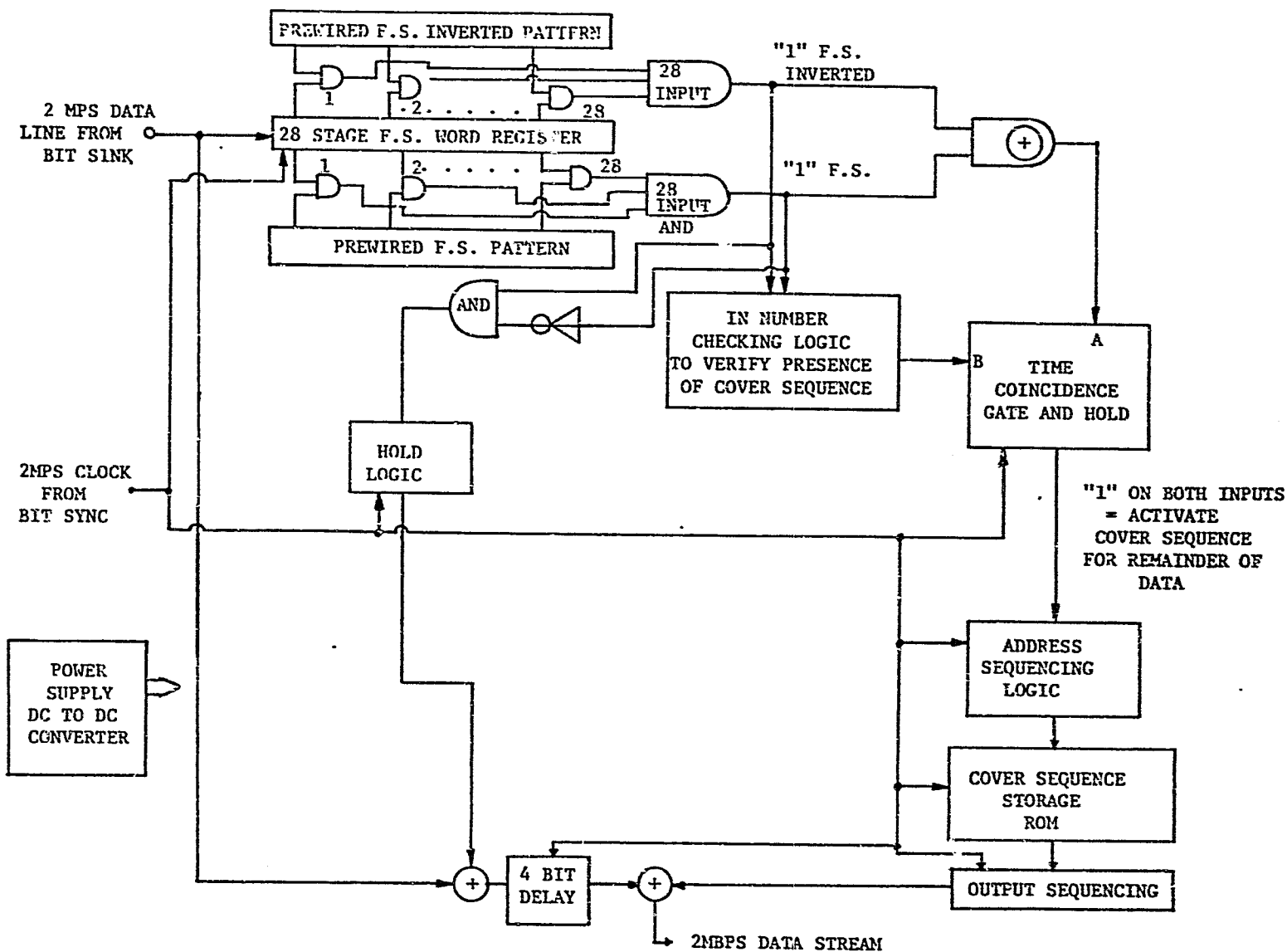


Figure B.4 Cover Sequence Decoder Block Diagram

ORIGINAL PAGE IS
OF POOR QUALITY

1. Frame synchronization is achieved by searching for two consecutive frame synchronization words. When the first word is recognized the 4 bit ID count is set in a counter. This count is then updated by 1 count and upon receipt of the next frame synchronization word the new 4 bit ID count is compared to the updated count and if there is agreement frame synchronization is achieved.

At this time a pulse is generated (from high to low) which lasts for one bit time during the 32nd bit of the frame synchronization word.

The data that is received during these two frame synchronization words is discarded.

2.0 Requirements

2.1 Performance

2.1.1 CSG Encoder

The CSG Encoder will reformat the 2 MBPS HRM serial bit stream with the following constraints.

- a. The probability of not having at least one transition in 64 bits shall be $5.4210108 \times 10^{-20}$ or less
- b. The probability of not having 64 transitions in 512 bits shall be $3.945^+ \times 10^{-71}$ or less
- c. The combined failure probability shall be 2.434×10^{-17} or less
- d. The reformatted data stream shall not increase or decrease the information rate
- e. The reformatted data shall be compatible with the existing BCH code
- f. The mechanization of the PN cover encoder shall have a minimal impact on the existing system
- g. Source voltage for the PN cover encoder will be provided by a 28 volt input DC to DC converter
- h. There shall be no increase in RF bandwidths
- i. The HRM sync word shall be unaffected by the CSG sequence
- j. There shall be no error propagation due to the use of the CSG

- k. The recommended 13th order polynomial sequence is

$$g(x) = 1 + x^9 + x^{10} + x^{12} + x^{13}$$
 which shall be truncated to a 3040 digit data stream
- l. The CSG shall pass through any data without an HRM frame sync undisturbed
- m. The CSG shall ignore any frame sync pulse signal that indicates a total data rate greater than 2 MBPS
- n. Parts for the CSG shall be IEEE.

2.1.2 CSG Decoder

The CSG Decoder will reformat the 2 MBPS HRM data streams with the following constraints:

- a. The CSG Decoder shall pass data without HRM frame sync (normal or complementary) undisturbed
- b. The CSG Decoder shall pass NRZ-S, NRZ-M and Bi- ϕ data undisturbed, assuming basic NRZ-L compatible logic levels
- c. The CSG Decoder shall have the capability to recognize both normal and complementary HRM frame sync. (If available the CSG Decoder may substitute a frame synchronization pulse from the HRDM in lieu of generating this information within the Decoder itself. The availability of this frame synchronization pulse from the HRDM must be resolved between the contractor and the con. acting NASA agency.)

APPENDIX C

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

A Monthly Progress Report
Covering the Period
November 1, 1980 - November 30, 1980

Submitted to:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama
35812

Submitted by:

Mississippi State University
Engineering and Industrial Research Station
Department of Electrical Engineering
Mississippi State, Mississippi
39762

Principal Investigator: Frank Ingels
Associate Investigator: William O. Schoggen

Contract No. NAS8-33887

A STUDY OF HIGH DENSITY BIT TRANSITION
REQUIREMENTS VERSUS THE EFFECTS ON BCH
ERROR CORRECTING CODING

Work Summary

Period: (November 1, 1980 to November 30, 1980)

A meeting was held at NASA-MSFC on November 10, 1980. Participants were Mr. David Mann (MSFC), Mr. Ellington Pitts (MSFC), Miss Virginia Johnson (MSFC), one representative from MDTSCO, Mr. W. O. Schoggen (MSU) and Dr. F. Ingels (MSU). The purpose of the meeting was to discuss the latest requirements on the system to be implemented for achieving a high density bit transition data stream for the 2 MBPS HRM formatted science data.

The requirements are:

For the Cover Sequence Generator (CSG) Encoder

1. Data input shall be NRZ-L from the HRM formatter. However, the encoder shall pass NRZ-S, NRZ-M or Bi- ϕ data streams in unaltered fashion as long as they have proper logic levels (that is logic levels compatible with the HRM NRZ-L data).
2. The encoder shall pass unaltered data streams of 2 MBPS which emanate from the system other than from the HRM formatter. These data streams will not contain a frame synchronization pattern similar to that of the HRM 2 MBPS formatted science data.
3. Only HRM 2 MBPS formatted data will be CSG encoded. If the HRM formatted data is being transmitted at a faster rate it shall not be encoded.

For the Cover Sequence Generator (CSG) Decoder

1. If no HRM frame synchronization (2 MBPS data rate) word occurs in the data stream from the bit synchronizer the data shall be passed on unaltered.
2. For HRM 2 MBPS formatted science data with proper frame synchronization word the data output shall be non-inverted in phase and the cover sequence removed.
3. The decoder shall pass NRZ-M, NRZ-S, or Bi- ϕ data without alteration. It is assumed that these data streams will have compatible logic levels with the HRM 2 MBPS formatted science data stream.

The CSG Encoder block diagram is illustrated in Figure C-1. The CSG Decoder block diagram is illustrated in Figure C.2.

In Figure 2 the Timing Gate and the Time Coincidence Gate serve the purpose of determining that the clock and frame sync are indeed a 2 MBPS rate rather than a higher rate. The hold portion of the Time Coincidence Gate is to activate the cover sequence generator for a major portion of a frame.

After recognition of a frame synchronization word the 4ID bits of the frame synchronization pattern are encoded to provide a 4 bit pattern for the decoder to use for determining that the cover sequence has actually been added to the data stream.

The design is fail safe in that the HRM 2 MBPS formatted data will pass through unaltered if the Encoder fails to activate the CSG, however, in this event the data stream is not guaranteed to contain a sufficient bit transition density.

In Figure C.2 the double frame synchronization word detectors suffice to determine if the data stream emanating from the bit synchronizer is in non-inverted or inverted phase. In the case of the inverted phase the data stream is reinverted automatically. The 4ID bits are checked for the presence of the cover sequence and if it is present and if the Time Coincidence Gate agrees that the data stream is a 2 MBPS rate then the cover sequence generator is activated.

The 4 Bit Delay is necessary for inspection of the 4ID bits prior to activation of the cover sequence generator.

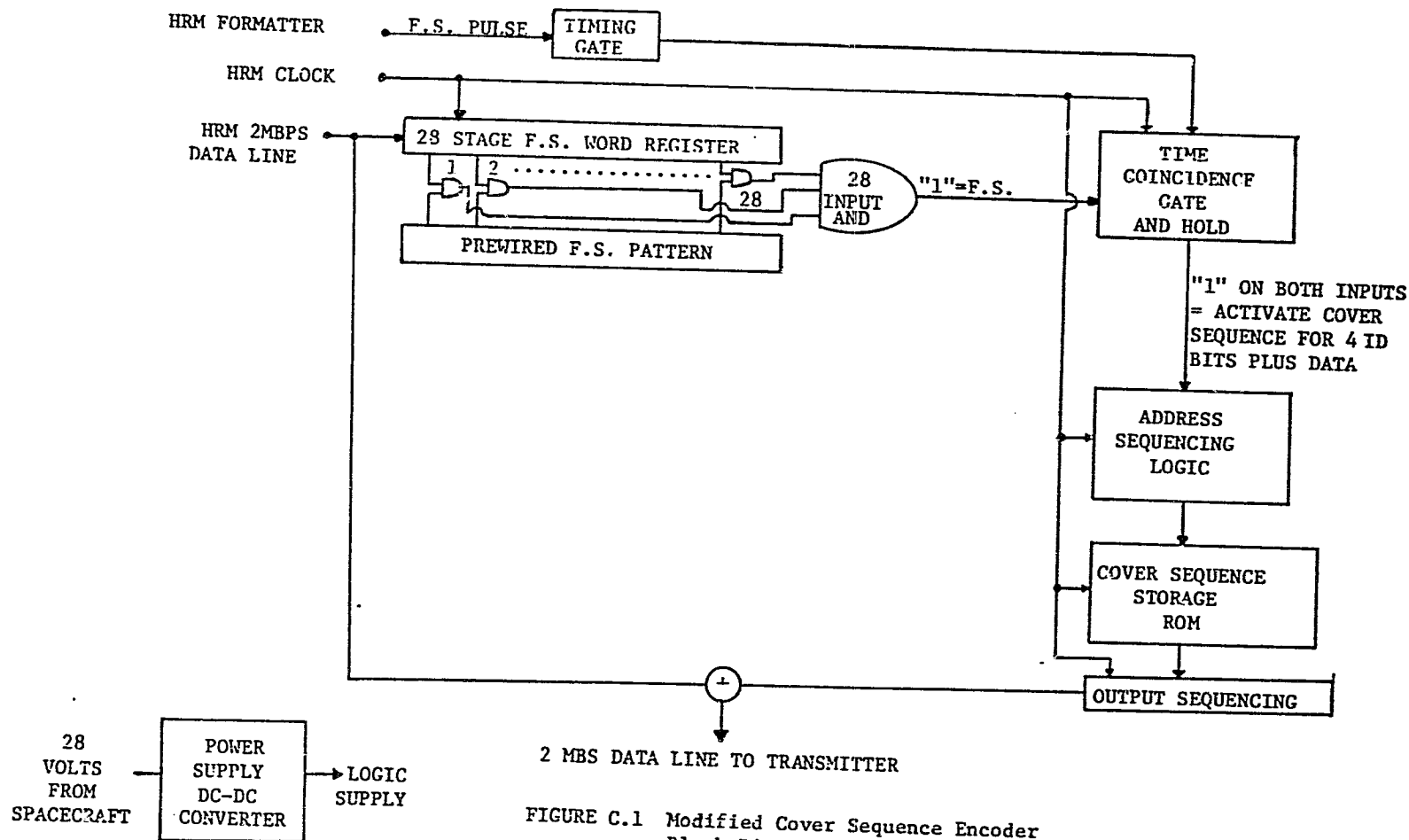


FIGURE C.1 Modified Cover Sequence Encoder
Block Diagram

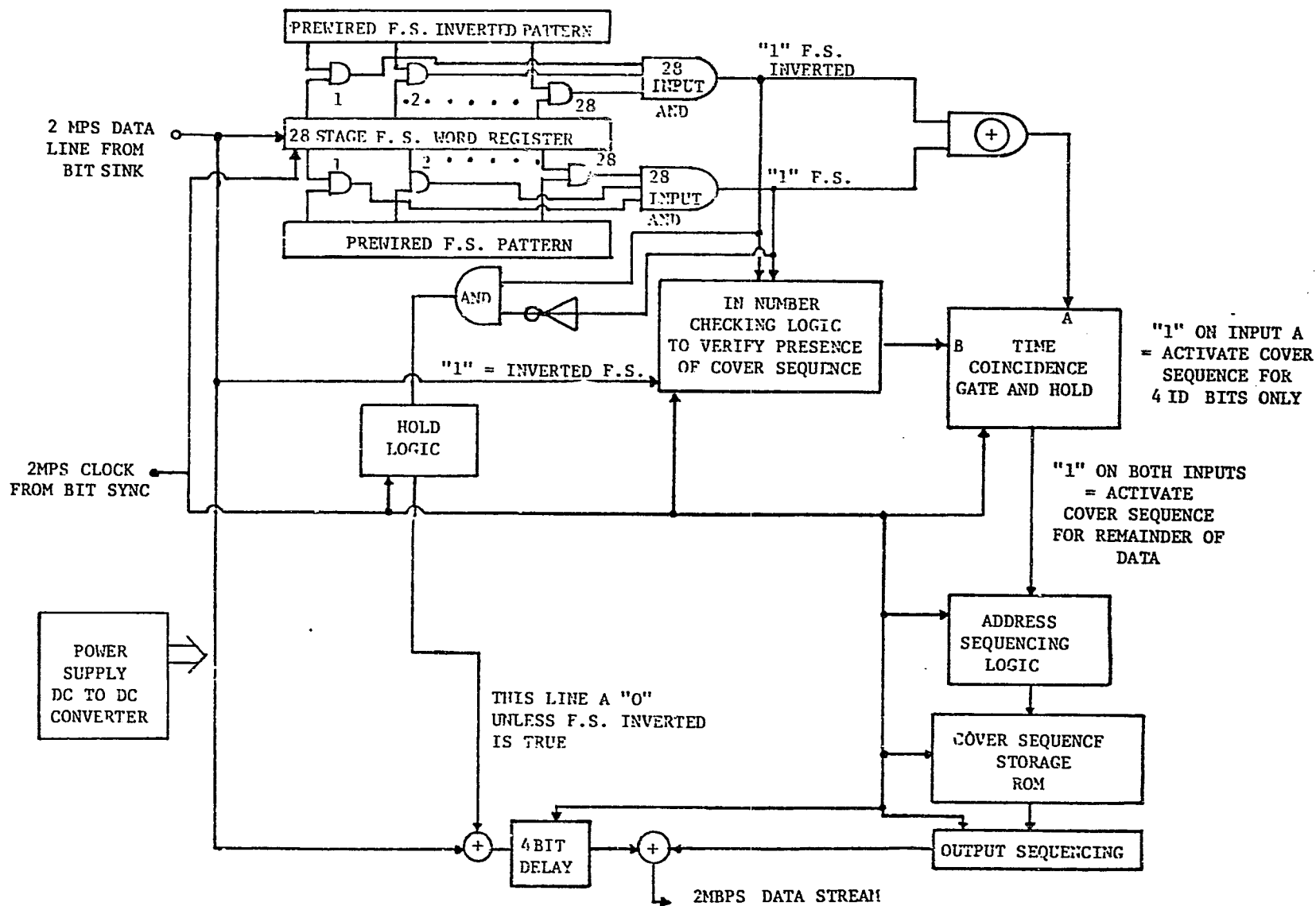


FIGURE C.2 Modified Cover Sequences Decoder Block Diagram